Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Sousse

Institut Supérieur des Sciences Appliquées et de Technologie

Introduction à l'Automatisme

GRAFCET & GEMMA

Cours 2ème année II

Mourad KCHAOU

mouradkchaou@gmail.com

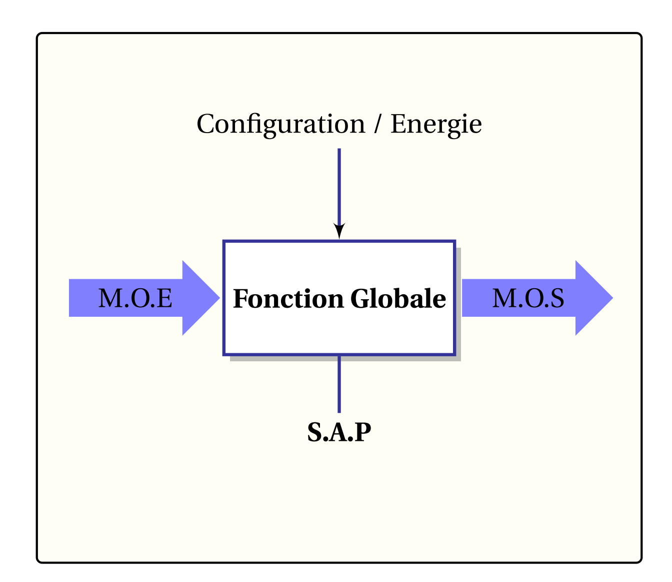
Table des matières

Chapitre 1

Introduction aux systèmes automatisés

1.1 Système de Production

Un Système de Production a pour rôle de conférer une valeur ajoutée à un ensemble de produits et matériaux bruts pour obtenir un produit fini ou un produit intermédiaire qui va, à son tour, participer à l'élaboration d'un produit fini. Le système de production reçoit donc un flux de matières d'œuvre (brutes), consomme divers types d'énergie (électrique, pneumatique, hydraulique, etc.) et consomme un certain nombre de composants auxiliaires (eau, liquide de refroidissement, huile, etc.). Il génère un flux de produits élaborés et un flux de déchets et de nuisances diverses.



1.1.1 Automatisation

Tout système de production nécessite la présence d'opérateurs humains d'exploitation, de réglage et de maintenance. Automatiser un système de production consiste donc à réduire l'intervention humaine tout au long du processus de fabrication et à optimiser l'utilisation des matières et des énergies par l'emploi de nouvelles technologies.

En d'autre terme, l'automatisation de la production consiste à transférer tout une ou partie des tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé **Partie Commande** (P.C).

La partie commande mémorise le *SAVOIR FAIRE* des opérateurs pour obtenir la suite des actions à effectuer sur les matières d'oeuvre afin d'élaborer la valeur ajoutée. Elle exploite un ensemble d'informations prélevées sur la **Partie Opérative** (P.O) pour élaborer la succession des ordres nécessaires pour obtenir les actions souhaitées.

Cette automatisation permet donc globalement de rendre plus compétitif le produit, l'entreprise et les hommes. Nous parlons alors de **Systèmes Automatisés de Production** (SAP).

1.1.2 Structure d'un système automatisé

Un Système Automatisé de Production est globalement composé d'une Partie Operative (P.O) et d'une Partie Commande (P.C) Figure 1.1.

La Partie Operative Elle est formée de l'ensemble des divers organes physiques qui interagissent sur le produit pour lui conférer une valeur ajoutée : les pré-actionneurs, les actionneurs et les capteurs.

- Les pré-actionneurs servent de relais de puissance entre la commande et les actionneurs.
- Les actionneurs transforment l'énergie reçue en une énergie exploitable par les effecteurs qui agissent et transforment le produit
- Les capteurs recueillent les informations : état ou position du produit, alarmes, etc. traduisant un changement d'état du procédé. Ces changements induisent le calcul de la commande du procédé de production par son Système de Contrôle/Commande ou Partie Commande.

La Partie Commande La Partie Commande (P.C) ou Système de Contrôle/Commande est l'ensemble des moyens logiciels et d'informations destiné :

- à coordonner la succession des actions sur la Partie Operative ;
- à surveiller son bon fonctionnement ;
- à gérer les dialogues avec les intervenants ;
- à gérer les communications avec d'autres systèmes ;
- à assurer le traitement des données et des résultats relatifs au procédé, aux matières d'œuvre, aux temps de production, à la consommation énergétique... (gestion technique).

Notons qu'elle comporte aussi une partie purement matérielle (câblage), imposée par les normes de sécurité.

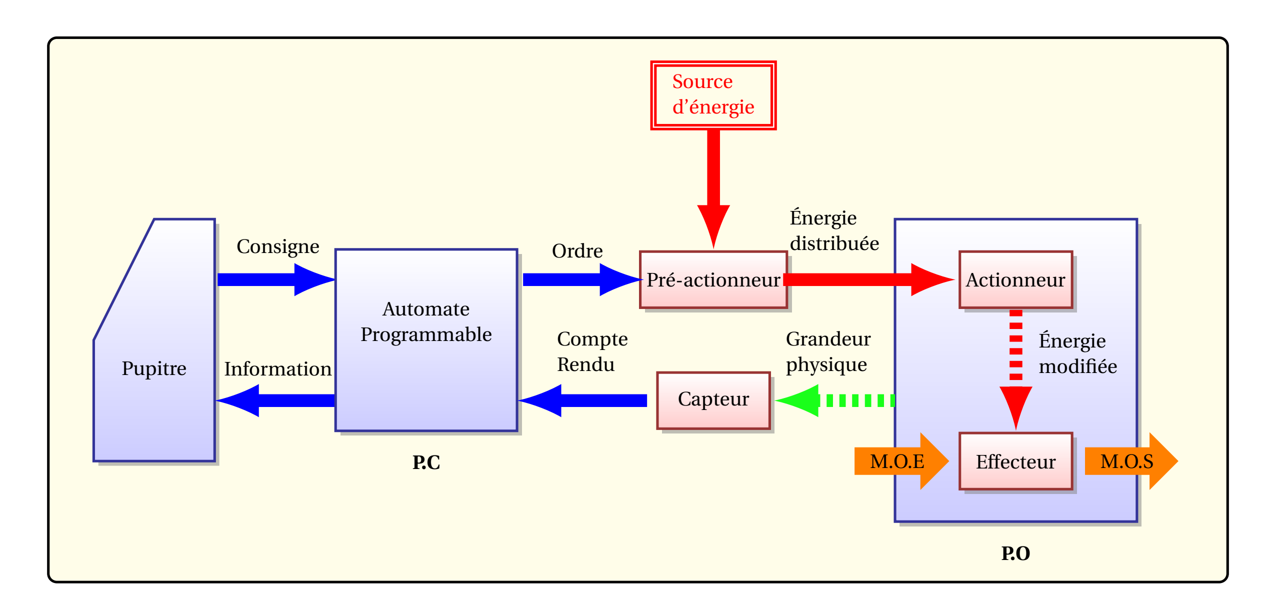


Figure 1.1: Structure d'un système automatisé

Un procédé industriel passe d'un état initial à un état final via une succession d'états intermédiaires qui marquent son évolution. Cette évolution est surveillée grâce aux capteurs qui renvoient des comptes-rendus de l'état du système. La Partie Commande traite ces informations et élabore les ordres renvoyés vers les actionneurs.

1.2 L'automate programmable industriel (API)

Les exigences qui sans cesse accrues de qualité, sécurité, flexibilité entraîne un accroissement des besoins, en particulier la manipulation d'un grand nombre de variables et la gestion de véritables flux de communication.

Cela explique que les systèmes câblés deviennent trop volumineux et trop rigides pour de telles applications, et que l'on se tourne donc vers des solutions utilisant les techniques de traitement de l'information par processeurs programmables.

1.2.1 Place des automates programmables

Dans ces systèmes de traitement de l'information, les API occupent une place de choix. Les équipements notés « commande » sont souvent des automates. Remplaçant initialement des ensembles en technologie câblée (relais électromagnétiques ou statiques, composants pneumatiques), ils constituent de plus en plus un maillon fiable et efficace entre le calculateur, qui a plutôt un rôle de gestion et l'appareillage de terrain (capteurs et actionneurs). Cet appareillage pouvant lui-même aujourd'hui contenir un processeur, il nous faut préciser la définition de l'API, tout en sachant que dans ce do-

maine comme dans beaucoup d'autres touchant à l'informatique, les frontières sont floues et mouvantes.

Nous considérerons comme automate programmable un système :

- construit autour d'un processeur numérique, spécifique ou non ;
- pouvant être relié à de nombreux signaux physiques ;
- fonctionnant grâce à une protection adaptée dans des conditions industrielles ;
- doté d'un logiciel de programmation permettant un traitement simple des variables booléennes (Tout ou Rien-TOR) ;
- doté de possibilités d'échanges avec d'autres processeurs.

À partir de ces définitions, nous distinguerons dans les fonctions que l'automate doit remplir :

- un rôle de commande où il est un composant d'automatisme, élaborant des actions, suivant une algorithmique appropriée, à partir des informations que lui fournissent des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques) ;
- un rôle de communication dans le cadre de la production :
  - avec des opérateurs humains : c'est le dialogue d'exploitation,
  - avec d'autres processeurs, hiérarchiquement supérieurs (calculateur de gestion de production), égaux (autres automates intervenant dans la même chaîne) ou inférieurs (instrumentation intelligente).

1.2.2 Structure matérielle

La structure matérielle interne d'un API obéit au schéma donné sur la Figure 1.2.

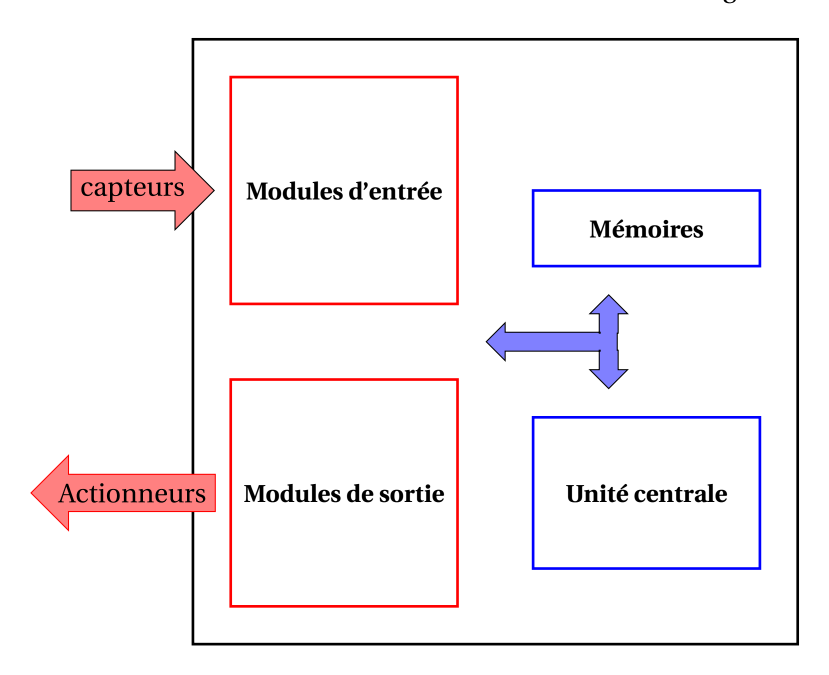


Figure 1.2: Architecture d'un API

1.2.2.1 Le processeur

Cœur de l'appareil, dans l'unité centrale, ce n'est sans doute pas, paradoxalement, le point le plus caractéristique, mais il conditionne tout de même largement les performances. L'unité centrale UC est une carte électronique bâtie autour d'un processeur, qui assure au moins les fonctions suivantes :

– opérations logiques sur bits, étant l'information élémentaire à deux états) ou sur mots (ensemble de bits, le plus souvent 16 pour les API),
– temporisation et comptage.

1.2.2.2 Modules d'entrées/sorties (E/S)

Ils assurent le rôle d'interface de la partie commande (PC) avec son environnement. Pour ce faire, ils doivent :

– regrouper des variables de même nature, pour diminuer complexité et coût ,
– assurer le dialogue avec l'UC ;
– « traduire » les signaux industriels en informations API et réciproquement, avec une protection de l'UC et un traitement adéquats.

Modules de base

Il s'agit principalement des entrées/sorties Tout ou Rien (E/S TOR), la gestion de ce type de variables constituant le point de départ historique des API et restant une de leurs activités majeures. Dans le cas de modules séparés, la modularité dépend des constructeurs, les valeurs 8 et 16 se rencontrant fréquemment. On trouve des modules pour tensions continues (24 V, 48 V) et alternatives (24 V, 48 V, 100/120 V, 220/240 V), les valeurs de ces tensions étant conformes à la norme EN 61131-2. Pour les automates à E/S intégrées, les caractéristiques des entrées et sorties sont choisies parmi les standards les plus répandus (entrées 24 V isolées, sorties transistorisées à alimentation continue, ou entrées 110 V alternatif isolées, sorties relais, par exemple, pour le TSX17).

La protection des circuits logiques d'entrée contre les parasites électriques est souvent résolue par découplage opto-électronique. Le passage des signaux par un stade de faisceau lumineux assure en effet une séparation entre les circuits électriques internes et externes.

Du côté sorties, on doit assurer le même type de protection, mais aussi une amplification de puissance, avec au final un courant continu ou alternatif selon les cas. Deux types principaux de cartes électroniques sont utilisés :

– les modules à sorties statiques (relais statiques intégrant des composants spécialisés : transistors bipolaires, IGBT, thyristors), sans usure mécanique et dont les caractéristiques de commutation se maintiennent dans le temps,

– les modules à relais électromagnétiques, où le découplage résulte de l'existence de deux circuits électriques (bobine, circuit d'excitation et contacts, circuits de puissance). D'une durée de vie plus limitée que les relais statiques (moins de 100 000 cycles pour les contacts soumis à 10 A sous 125 V alternatif) et plus lents, les relais électromagnétiques ont aussi des avantages : faible résistance de contact, faible capacité de sortie, faible coût.

Modules spécialisés

Nous rangerons dans cette catégorie des cartes qui assurent non seulement une liaison avec le monde extérieur mais aussi une partie du traitement pour soulager le processeur et donc améliorer les performances. De tels modules comportent donc un processeur spécifique ou une électronique spécialisée. Nous pouvons citer les suivants.

Cartes de comptage rapide elles permettent de saisir des événements plus courts que la durée du cycle , travaillant à des fréquences qui peuvent dépasser 10 kHz.

Cartes d'acquisition sous cette appellation nous rangerons non seulement les acquisitions de mesures (entrées analogiques), mais aussi les sorties analogiques ; il existe des modules mixtes regroupant entrées et sorties. Dans les deux cas, il faut une conversion entre les valeurs analogiques et les mots manipulables par l'API. La valeur analogique correspond à un nombre de bits (8 à 16) tel que l'erreur de quantification correspondante s'avère généralement inférieure à la résolution des capteurs ou à l'effet de seuil sur la commande des actionneurs. Les grandeurs analogiques reçues ou fournies obéissent à des standards électriques dont les plus fréquemment rencontrés sont le 4-20 mA, transmission en courant qui évite l'affaiblissement du signal.

1.2.2.3 Éléments de stockage et de liaison

Le stockage des données et des programmes s'effectue dans des mémoires. La mémoire vive (RAM) est volatile mais secourue par batterie. La mémoire morte (ROM) dont l'utilisateur ne peut que lire le contenu, éventuellement programmable (PROM) à l'aide d'outils spéciaux, contient le système d'exploitation, tandis que les programmes au point et utilisables peuvent se stocker dans des mémoires reprogrammables (EPROM), là encore avec un matériel spécifique. La mémoire de données contient elle-même plusieurs zones :

– une zone de bits, dont certains peuvent être secourus en cas de défaillance de l'alimentation électrique ;
– une zone de mots, permettant de soumettre des données à un traitement plus large que le traitement booléen (traitement numérique ou alphanumérique).

1.3 Programmation de l'API

La norme IEC 1131-3 définit entre autres choses, cinq langages qui peuvent être utilisés pour la programmation d'applications d'automatisme. Les cinq langages sont :

SFC (« sequential function char ») issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels ;

FBD (« function block diagram », ou schéma par blocs) ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables ;

LD (« ladder diagram », ou schéma à relais) ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true/false) ;

ST (« structured text » ou texte structuré) ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe ;

IL (« instruction list », ou liste d'instructions) ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.

1.3.1 C'est quoi un programme ?

Un programme est une unité logique de programmation qui décrit des opérations entre les variables de l'application. Un programme décrit des opérations séquentielles ou cycliques. Un programme cyclique est exécuté systématiquement à chaque cycle automate. L'exécution d'un programme séquentiel respecte les règles d'évolution du langage SFC.

1.3.2 Langage Ladder

Le langage LD (ladder diagram) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts.

L'idée initiale du Ladder est la représentation de fonction logique sous la forme de schémas électriques. Cette représentation est originalement matérielle : quand l' Automate Programmable Industriel n'existait pas, les fonctions étaient réalisées par des câblages. Par exemple, pour réaliser un ET logique avec des interrupteurs, il suffit de les mettre en série. Pour réaliser un OU logique, il faut les mettre en parallèle. Partant de ces principes, le Ladder a été créé et normalisé dans la norme CEI 61131-3. Il est, depuis, très utilisé dans la programmation des Automates Programmables Industriels.

Un programme Ladder se lit de haut en bas et l'évaluation des valeurs se fait de gauche à droite. Les valeurs correspondent en fait, si on le compare à un schéma électrique, à la présence ou non d'un

potentiel électrique a chaque nœud de connexion. En effet, le Ladder est basé sur le principe d'une alimentation en tension représentée par deux traits verticaux reliée horizontalement par des bobines, des contacts et des blocs fonctionnels, d'où le nom 'Ladder' (échelle).

Les composants du langage

Il existe 3 types d'élément de langage :

– les entrées (ou contacts), qui permettent de lire la valeur d'une variable booléenne. Figure 1.4

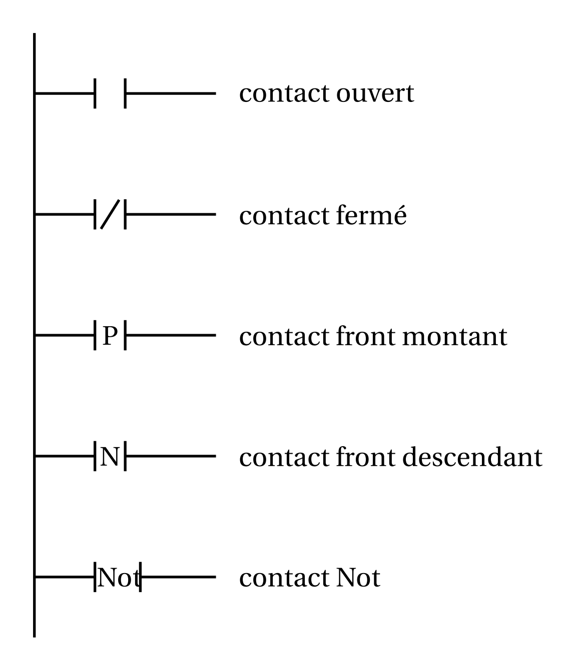


Figure 1.3: Les contacts

– les sorties (ou bobines) qui permettent d'écrire la valeur d'une variable booléenne. Figure 1.5

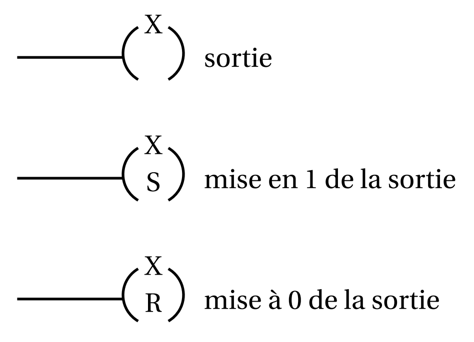


Figure 1.4: Opérations de sortie

– les blocs fonctionnels qui permettent de réaliser des fonctions avancées (temporisation, comptage, etc).

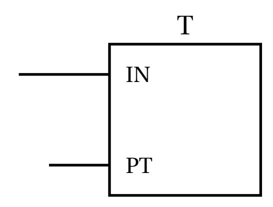


Figure 1.5: Temporisation

L’ opération *Démarrer temporisation* sous forme de retard à la montée s’écoule jusqu’à la valeur maximale lorsqu’elles est activée. Lorsque la valeur en cours « T » est supérieure ou égale à la valeur prédéfinie PT, le bit de temporisation T est activé. La temporisation « retard à la montée » est remise à zéro lors de sa désactivation. Ce temporisation s’arrête lorsque sa valeur maximale est atteinte.

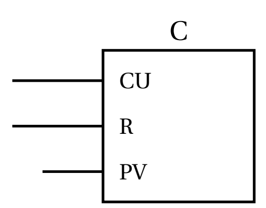


Figure 1.6: Compteur incrémental

L’opération Compteur incremental incrémente jusqu’à la valeur maximale en cas de front montant à l’entrée d’incrémentation CU. Lorsque la valeur en cours « C » est supérieure ou égale à la valeur prédéfinie PV, le bit de compteur C est activé. Le compteur est remis à zéro lorsque l’entrée de remise à zéro R est activée.

Chapitre 2

Le GRAFCET

2.1 Définition

Le langage GRAFCET (**GRA**phe **F**onctionnel de Commandes **É**tapes **T**ransitions) a été introduit en 1977 par l'AFCET (Association Française pour la Cybernétique Économique et Technique.
C'est un *modèle* permettant de *spécifier* et *concevoir* un **automatisme séquentiel**. C'est un *outil graphique*, puissant permettant de définir le comportement **séquentiel** d'un système automatisé à partir de la connaissance des actions à entreprendre, associées à des variables de SORTIES, et des événements qui peuvent permettre le passage d'une situation à une autre, associés à des variables d'ENTRÉE

Le GRAFCET peut être utilisé pour représenter l'automatisme dans toutes les phases de la conception : de la définition du cahier des charges, la mise en oeuvre (programmation d'un automate programmable industriel, utilisation de séquenceurs ou autres technologies) en passant par l'étude des modes de marches et d'arrêts.
Le GRAFCET repose sur l'utilisation d'instructions précises, l'emploi d'un vocabulaire bien défini, le respect d'une syntaxe rigoureuse, et l'utilisation de règles d'évolution. Il permet, entre autre, d'adopter une démarche progressive dans l'élaboration de l'automatisme.

2.1.1 Points de vue

La description du comportement attendu d'une partie commande peut se représenter par un GRAFCET d'un certain « niveau ». La caractérisation du «niveau» du GRAFCET nécessite de prendre en compte trois dimensions :

- une dimension « point de vue », caractérisant le point de vue selon lequel un observateur s'implique dans le fonctionnement du système pour en donner une description. On distingue trois points de vue :
  - un point de vue « système », ou « procédé »,
  - un point de vue « partie operative »,
  - un point de vue « partie commande » ou « réalisateur ».

– une dimension « spécifications», caractérisant la nature des spécifications techniques auxquelles doit satisfaire la partie commande. On distingue trois groupes de spécifications :
  – spécifications fonctionnelles,
  – spécifications technologiques,
  – spécifications opérationnelles.
– une dimension « finesse », caractérisant le niveau de détail dans la description du fonctionnement, d'un niveau global (ou macro-représentation) jusqu'au niveau de détail complet où toutes les actions et informations élémentaires sont prises en compte.

2.2 Structure graphique

Un GRAFCET est un graphe composé de **séquences**, c'est-à dire une *succession* d'**étapes** et de **transitions** reliées par des **arcs orientés**.

Exemple

Pour percer une pièce, la séquence suivante s'exécute après une action de l'opérateur de mise en marche :

- serrer la pièce,
- Percer la pièce,
- desserrer la pièce.

Il s'agit d'un fonctionnement séquentiel qui peut être représenté par un GRAFCET.

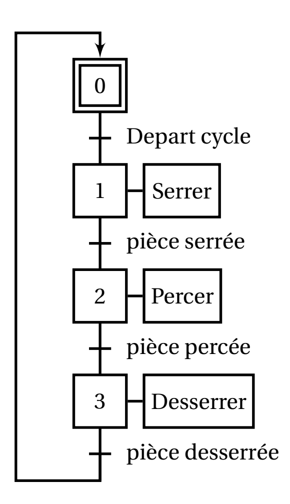


Figure 2.1: Structure d'un GRAFCET

2.2.1 Étapes-Actions

Les étapes

- une ***étape*** caractérise un comportement invariant d'une partie ou de la totalité de la partie commande ; à un instant donné et suivant l'évolution du système,
- une ***étape*** est soit ***ACTIVE***[a] ou ***INACTIVE***,
- l'ensemble des *étapes actives* définit la ***SITUATION*** de la partie commande,
- les étapes qui sont actives au début du processus sont appelées ***étapes initiales*** qui seront représentées par un *carré double*,
- une étape est repérée numériquement par un numéro, i, et possède ainsi une variable d'état notée X_i
  - $X_i = 1$ quand l'étape est active.
  - $X_i = 0$ quand l'étape est inactive.

a. Repérée par ∗ pour expliquer le fonctionnement.

Les actions

- une ou plusieurs **actions** peuvent être associées à une étape
- une ***action*** traduit **« ce qui doit être fait»** chaque fois que l'étape à laquelle elle est associée est active.

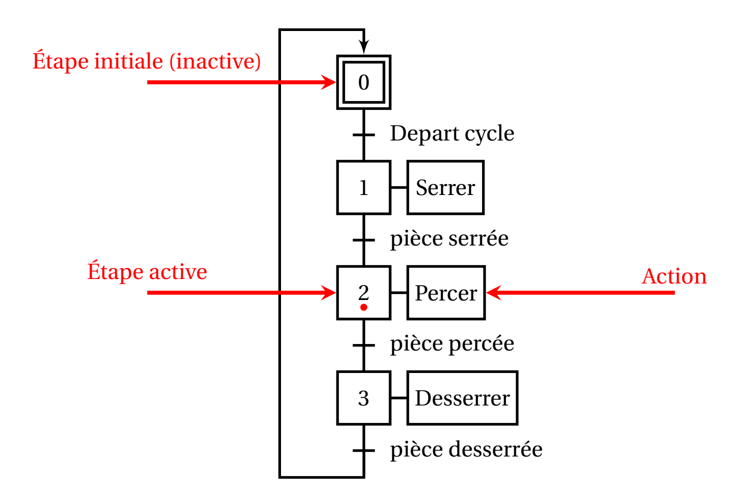


Figure 2.2: Étapes-Actions

2.2.2 Transitions-Réceptivités-Liaisons orientées

Les transitions

- une ***transition*** est représentée par un petit trait perpendiculaire à la liaisons orientée,
- une ***transition*** indique la possibilité d'évolution entre plusieurs étapes,
- une ***transition*** peut être **franchie** provoquant le changement d'activité des étapes.

Les réceptivités

- une ***réceptivité*** est une **condition logique** associée à une transition,
- une ***réceptivité*** peut être ***vraie*** ou ***fausse***.

Les liaisons orientées

- une liaison orientée relie les étapes aux transitions et les transitions aux étapes,
- une liaison orientée indique les voies d'évolution.

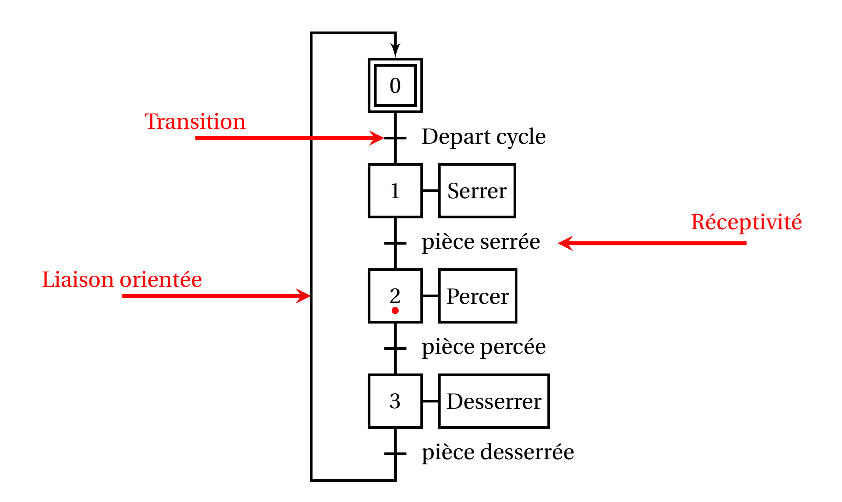
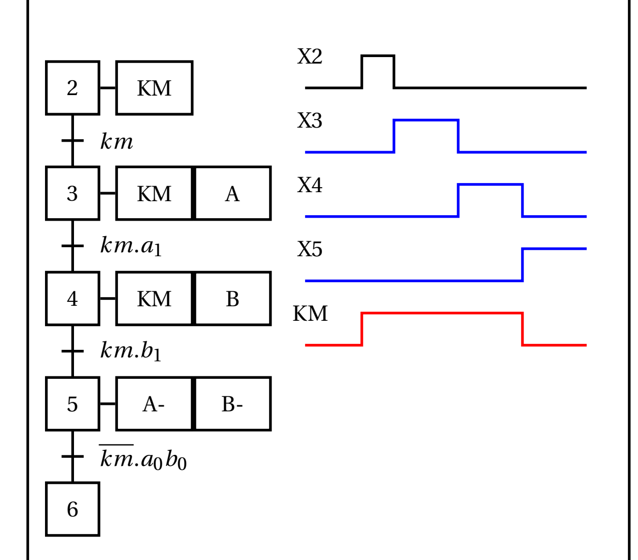
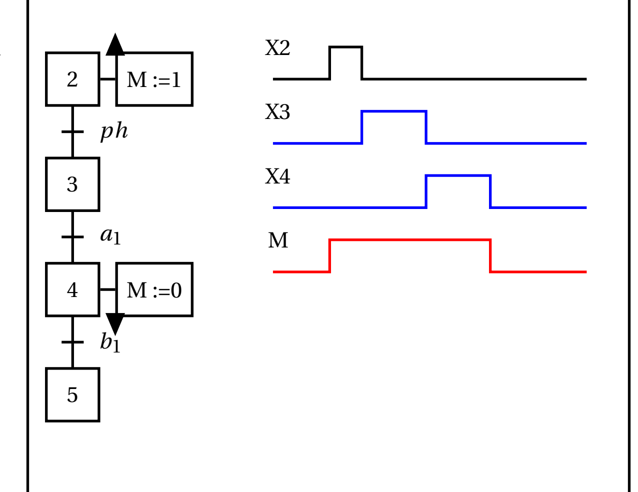


Figure 2.3: Transitions-Réceptivités-Liaisons orientées

2.3 Description détaillée d'actions

Le critère retenu pour la classification des actions les plus usitées est la durée de l'action comparativement à la durée d'activité d'une étape.

Action	Symbole/Chronogramme
Action continue : L'exécution de l'action continue se poursuit tant que l'étape à laquelle elle est associée reste active. Plusieurs actions continues peuvent être associées à une étape. Le symbole d'action continue contient le libellé d'assignation	2 A X2 A
Action conditionnelle : C'est une action continue dont l'exécution est soumise à une condition logique. La condition d'assignation peut être une expression boolèenne.	h 2 A X2 h A
Action temporisée : C'est un cas particulier d'action conditionnelle dans laquelle le temps intervient comme condition associée à l'action. *Action retardée :* la condition d'assignation n'est vraie qu'après une durée t_1 depuis l'activation de l'étape. *Action limitée dans le temps :* la condition d'assignation n'est vraie que pendant une durée t_1 depuis l'activation de l'étape.	t_1/X_2 2 A X2 A t_1 $\overline{t_1/X_2}$ 2 A X2 A t_1

Action à effet maintenu : L'action doit se poursuivre pendant la durée d'un certain nombre d'étapes consécutives : Effet maintenu par des actions continues non mémorisées ; – soit répéter l'action dans toutes les étapes, – soit utiliser les structures de séquences simultanées.	
Action mémorisée à l'activation et la désactivation : Une action mémorisée décrit comment affecter une valeur à une sortie qui la conserve. L'action mémorisée doit obligatoirement être associée à un événement interne (activation, désactivation, événement ou franchissement). A l'initialisation, la valeur de cette sortie est nulle.	

Actions manuelles

une action manuelle est une action particulière qui correspond à une intervention de l'opérateur (chargement, mise en position de la pièce, etc.) ne peut être considérée comme une action associée à une étape car elle ne correspond pas à un ordre issu de la partie commande vers la partie opérative. Dans ce cas, l'étape doit être considérée comme une étape sans action associée (étape vide). Si l'action est spécifiée à côté de l'étape, elle ne doit pas être encadrée mais signalée comme un commentaire entre guillemets.

L'activation de l'étape correspond alors à l'attente d'un événement extérieur et seul le compte-rendu de cet événement reçu par la partie commande permet de faire évoluer le système (Ex : signal pièce en place).

2.4 Description détaillée de réceptivités

Les variables :

Une variable **externe** est soit :

- Une variable binaire délivrée par la partie opérative à commander (état des capteurs) ou par son environnement (état d'un bouton manipulé par l'opérateur),
- Une variable binaire relative au temps.

Une variable **interne** est soit :

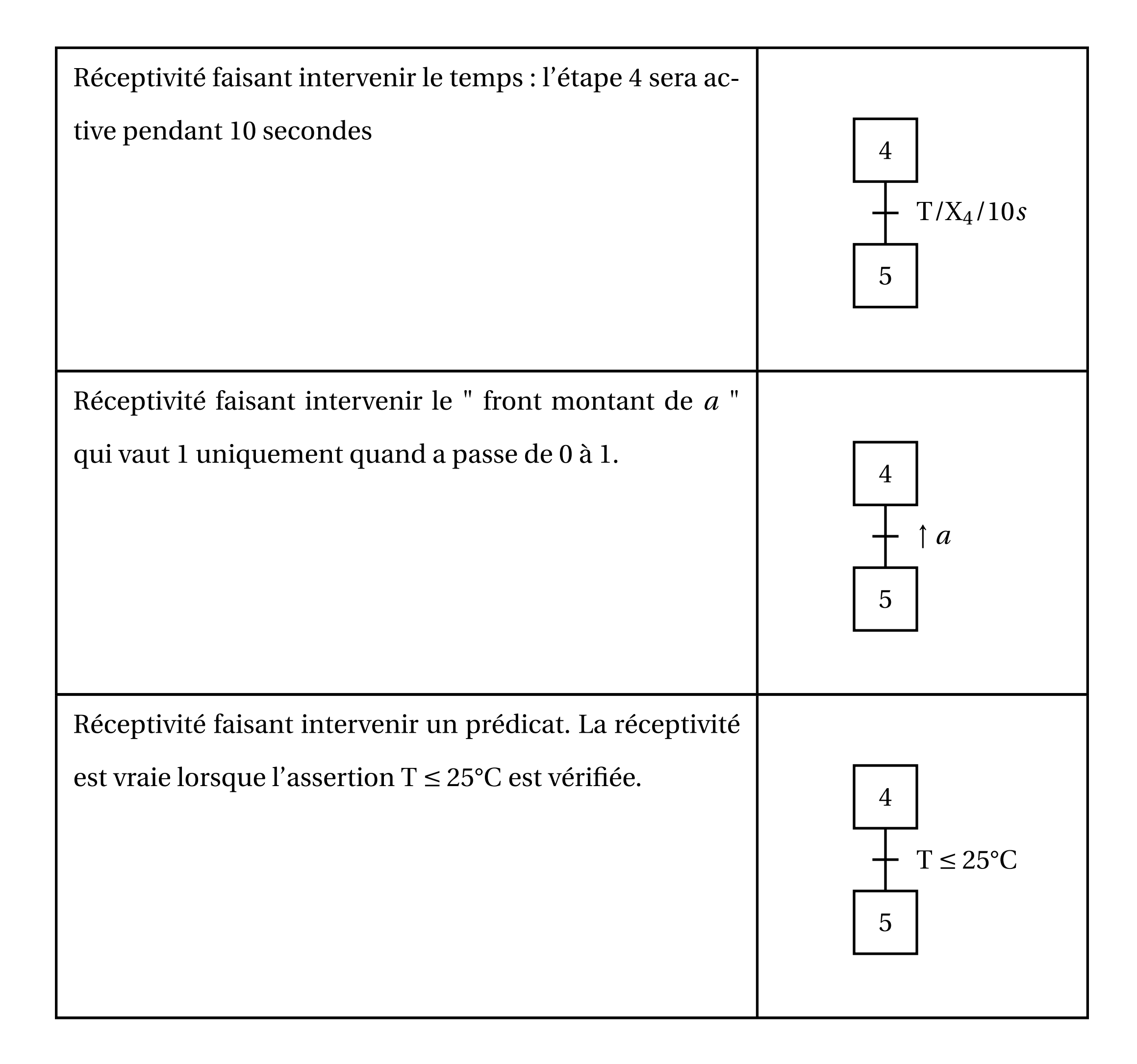
- Une variable binaire relative à la partie commande, c'est à dire à la situation dans laquelle se trouve le GRAFCET.
- Une variable générée par le modèle GRAFCET (compteur, variable de calcul...).
- Une variable binaire relative à un prédicat (ex : [T > 8°C] signifie que lorsque la proposition logique «T > 8°C » sera vérifiée alors la variable binaire [T > 8°C] sera égale à 1).

Les événements :

Un événement est un changement d'état d'une variable binaire.

- On notera $\uparrow a$ l'événement d'entrée front montant de "a" associé au passage de la valeur 0 à la valeur 1 de la variable a.
- On notera $\downarrow a$ l'événement d'entrée front descendant de "a" associé au passage de la valeur 1 à la valeur 0 de la variable a.

Type de réceptivité	Exemple
Réceptivité inscrite de façon littérale	4 — Position atteinte — 5
Réceptivité inscrite de façon symbolique à la droite de la transition	4 — $(a+\overline{b})c$ — 5



Réceptivité faisant intervenir le temps : l'étape 4 sera active pendant 10 secondes	4 — $T/X_4/10s$ — 5
Réceptivité faisant intervenir le " front montant de a " qui vaut 1 uniquement quand a passe de 0 à 1.	4 — $\uparrow a$ — 5
Réceptivité faisant intervenir un prédicat. La réceptivité est vraie lorsque l'assertion T ≤ 25°C est vérifiée.	4 — T ≤ 25°C — 5

2.5 Les structures de base du GRAFCET

Pour un GRAFCET, on distingue trois structures :

2.5.1 GRAFCET à séquence unique

GRAFCET à séquence unique

- une séquence est tout ensemble d'étapes successives où chaque étape est suivie d'une seule transition et chaque transition n'est validée que par une seule étape.
- Le GRAFCET le plus simple est le GRAFCET linéaire qui ne comporte qu'une seule séquence.

Exemple

La Figure 2.4 montre un chariot CH_1 pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite entre deux positions a_1 et b_1. Initialement, le chariot est au repos en a_1. Lorsqu'on appuie sur bouton dcy, il part vers la droite puis il retourne à gauche après une attente de $30s$.

– Établir le GRAFCET correspondant au fonctionnement du système.

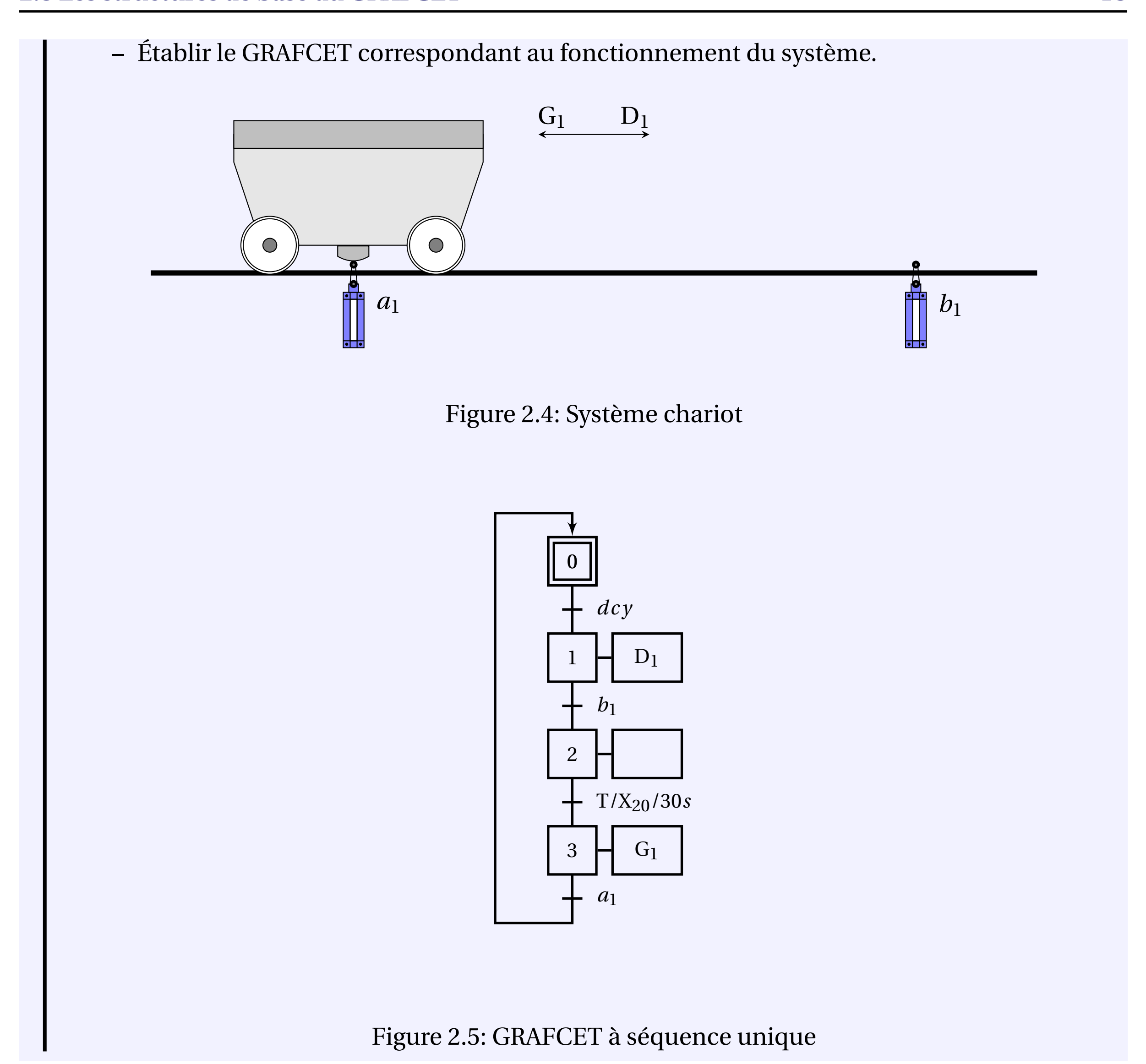


Figure 2.4: Système chariot

Figure 2.5: GRAFCET à séquence unique

2.5.2 GRAFCET à selection de séquences

GRAFCET à selection de séquences

- la sélection de séquences permet de faire un ***choix*** d'une séquence à partir d'une étape,
- la sélection de séquences est composée d'une **divergence en OU**, symbolisée par un trait de liaison simple, qui relie plusieurs transitions comportant les conditions de choix,
- chaque transition est suivie d'une séquence et une **convergence en OU** qui termine la sélection.
- le franchissement de chaque transition conduira à la réalisation de la séquence ou de l'action qui suit cette transition,
- le choix entre les séquences doit être **exclusif**, c'est-à-dire que seulement une des transitions peut être franchie à la fois.

Exemple

La Figure 2.6 montre un chariot CH_1 pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite. Initialement, le chariot est au repos en a_1. Si l'on appuie sur bouton S_1, il part jusqu'à $b1$ puis en appuyant sur r_1 il revient en a_1. Si l'on appuie sur bouton S_2, il part jusqu'à $b2$ puis il revient après une attente de 30s.

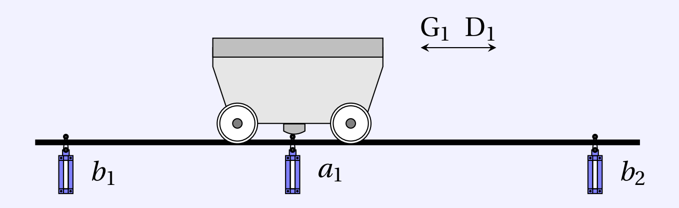


Figure 2.6: Système chariot

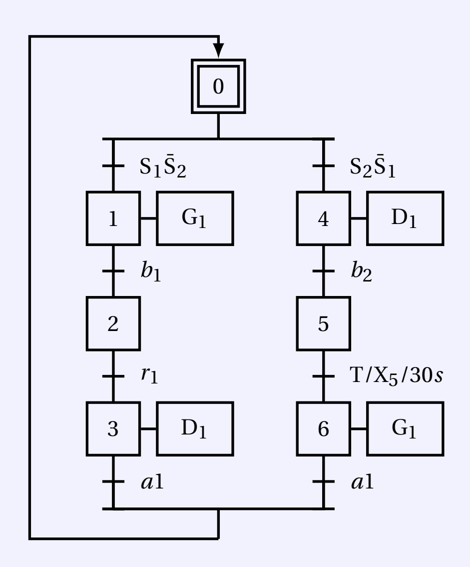


Figure 2.7: GRAFCET à sélection de séquences

2.5.2.1 Saut d'étapes

Saut d'étapes

- Le **saut d'étapes** est un cas particulier de la sélection de séquences,
- Le **saut d'étapes** permet soit d'exécuter une séquence, soit de sauter des étapes. Une des branches de la sélection ne contient donc aucune étape.

Exemple

La figure 2.8 représente un chariot CH_1 pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite. Le chariot peut faire deux cycles différents à partir de la position de repos en a_1 :

– En sélectionnant le commutateur C_1, il part vers b_1, puis vers b_2 après 30s d'attente et enfin il revient immédiatement à a_1.
– En sélectionnant le commutateur C_2 , il part directement vers vers b_2 enfin il revient immédiatement à a_1.

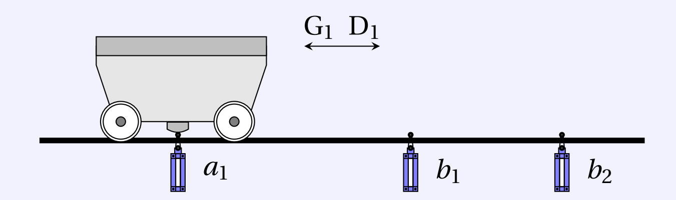


Figure 2.8: Système chariot

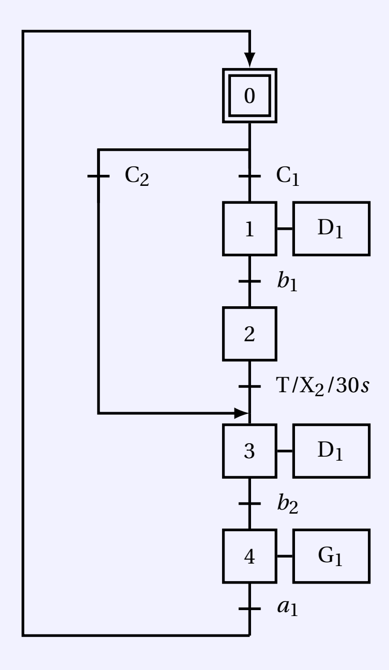


Figure 2.9: Saut d'étapes

2.5.2.2 Reprise d'étapes

Reprise d'étapes

La reprise d'étapes est utilisée lorsqu'il faut répéter une même séquence jusqu'à ce qu'une condition soit satisfaite.

Exemple

La figure 2.8 représente un chariot CH_1 pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite. En appuyant sur le bouton départ cycle dcy, le chariot se déplace vers b_1 où il s'arrête pendant 30s puis vers b_2 où il attend pendant 50s avant de retourner gauche vers a_1.

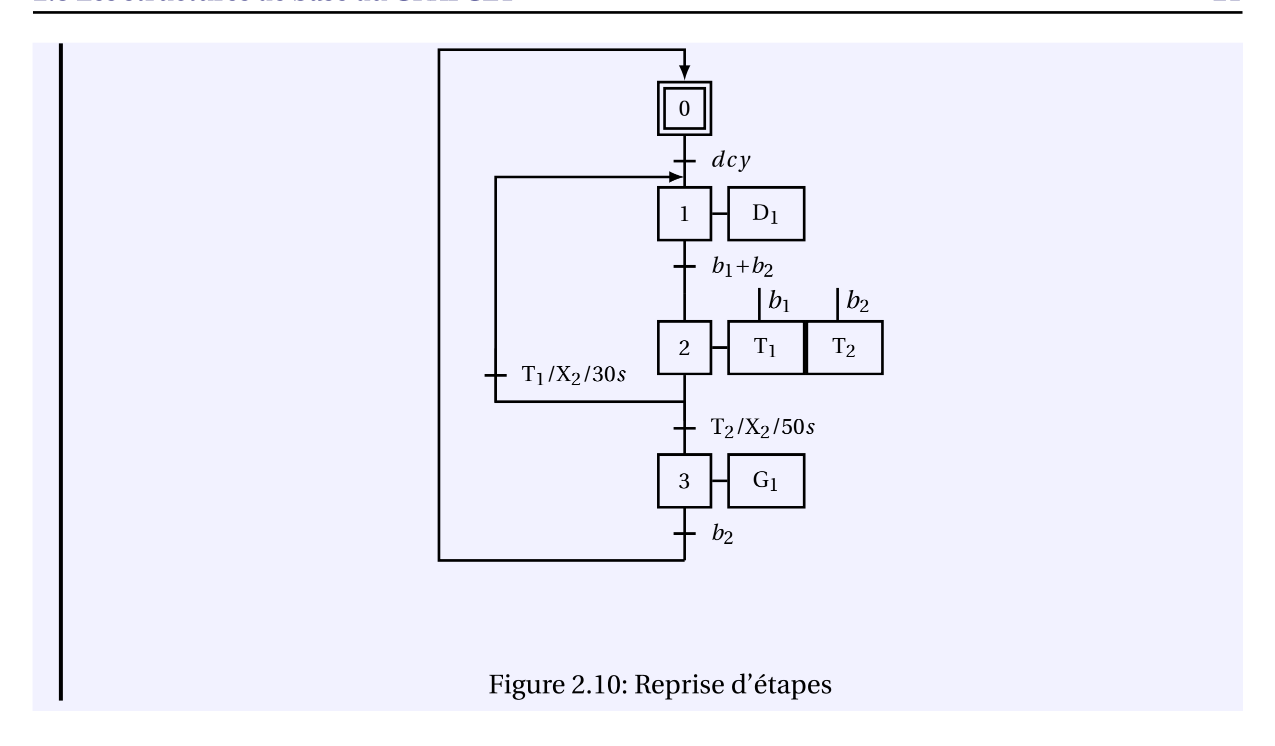
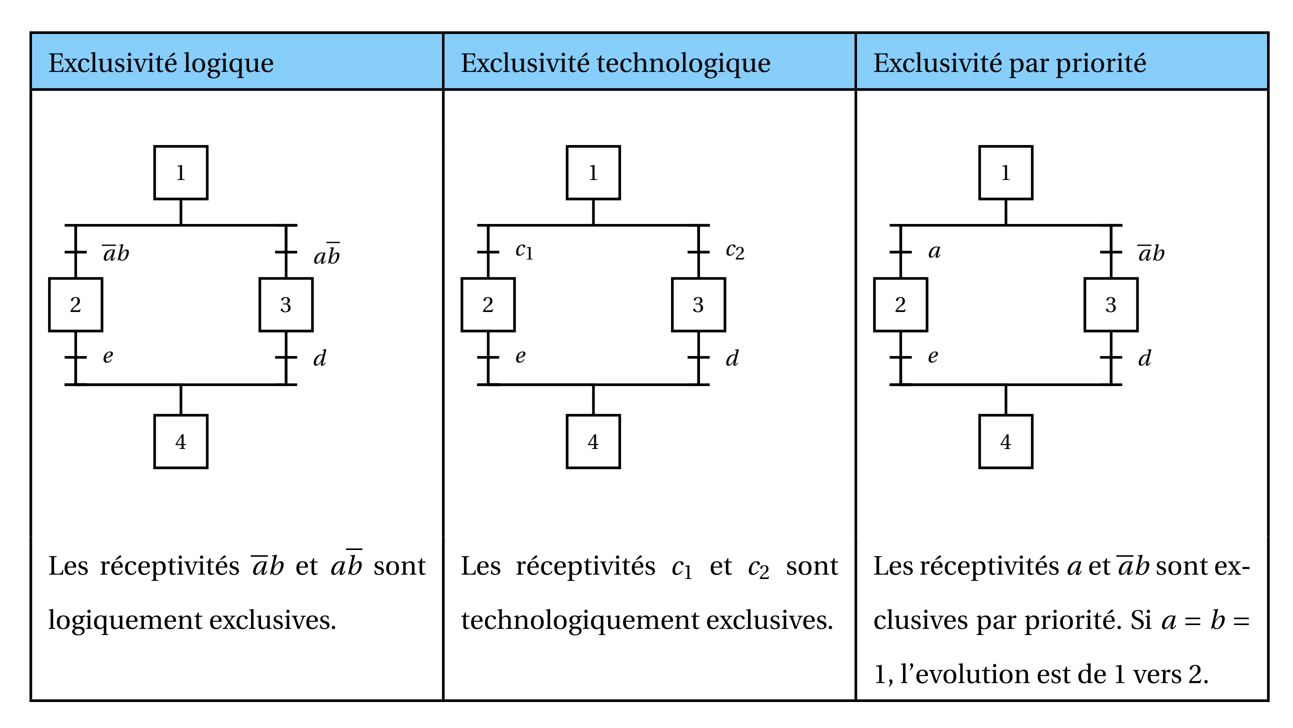


Figure 2.10: Reprise d'étapes

2.5.2.3 Types d'exclusivités

On distingue trois types d'exclusivité :



Exclusivité logique	Exclusivité technologique	Exclusivité par priorité
Les réceptivités $\overline{a}b$ et $a\overline{b}$ sont logiquement exclusives.	Les réceptivités c_1 et c_2 sont technologiquement exclusives.	Les réceptivités a et $\overline{a}b$ sont exclusives par priorité. Si $a = b = 1$, l'evolution est de 1 vers 2.

Tableau 2.3: Type d'exclusivité

2.5.3 GRAFCET à séquences simultanées

GRAFCET à séquences simultanées

- le **parallélisme** est l'exécution simultanée de plusieurs séquences, il est réalisé par une transition qui possède plusieurs étapes de sortie.
- l'exécution en parallèle de plusieurs séquences débute par une **divergence en ET**, représentée par un trait de liaison double.
- les séquences parallèles débutent simultanément mais l'évolution dans chaque branche est indépendante. Les séquences parallèles se terminent par une **synchronisation**,
- la synchronisation consiste en une **convergence en ET**, symbolisée par un trait de liaison double, qui est une transition possédant plusieurs étapes d'entrée,
- la transition qui suit la synchronisation n'est validée que lorsque toutes les étapes d'entrée sont actives.

Exemple

On considère deux chariots CH_1 et CH_2 pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite entre deux positions a_1 et b_1 pour CH_1 et a_2 et b_2 pour CH_2 figure (2.11). Les chariots sont tous d'eux au repos (en a_1 et a_2), et lorsqu'on appuie sur le bouton dcy, ils partent simultanément vers la droite. Le premier chariot qui arrive à l'extrémité droite repart à gauche et provoque le retour de l'autre chariot. Un nouveau départ ne peut être donné que s'ils sont tous d'eux en position gauche.

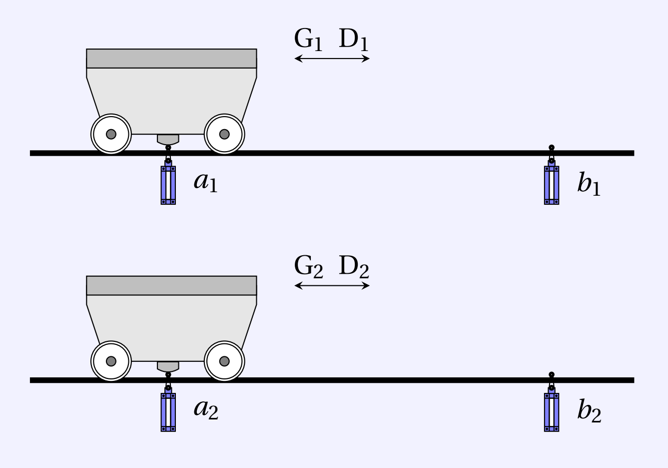


Figure 2.11: Deux chariots

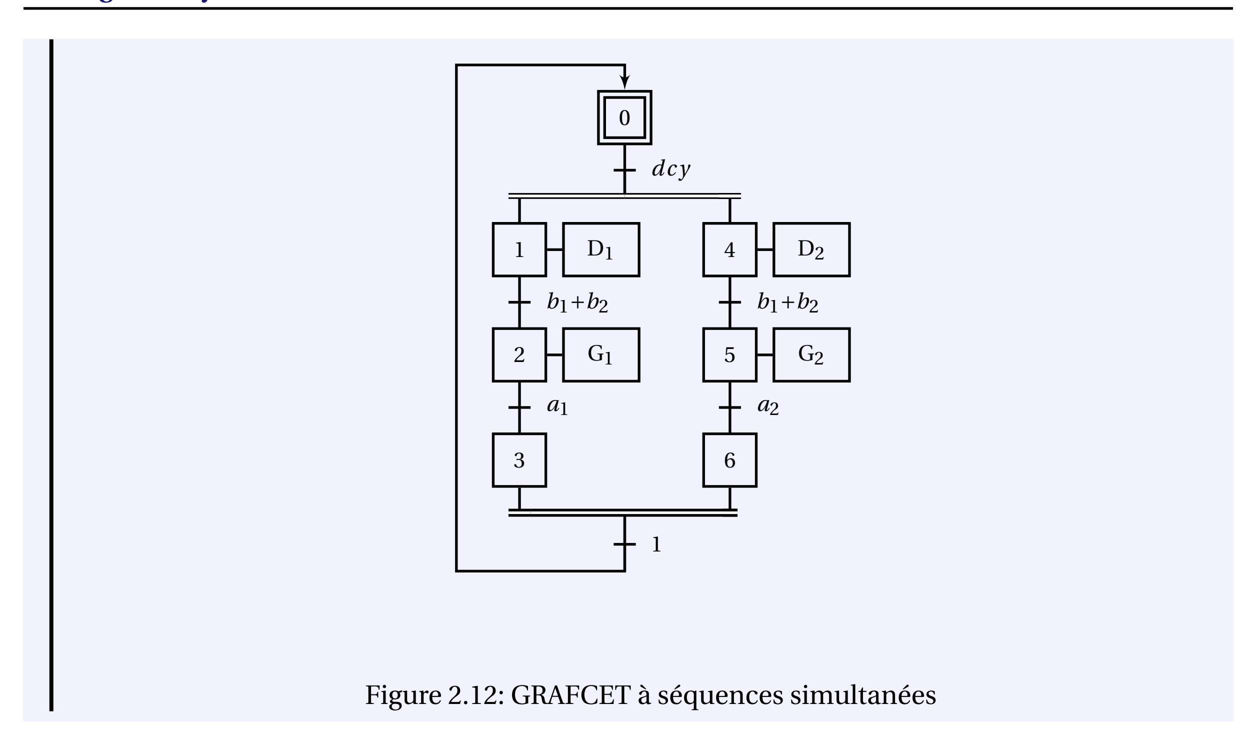


Figure 2.12: GRAFCET à séquences simultanées

2.6 Règles de syntaxe

Règles de syntaxe

- l'alternance étape-transition et transition-étape doit toujours être respectée quelle que soit la séquence parcourue,
- deux étapes ou deux transitions ne doivent jamais être reliées par une liaison orientée. La liaison orientée relie obligatoirement une étape à une transition ou une transition à une étape.

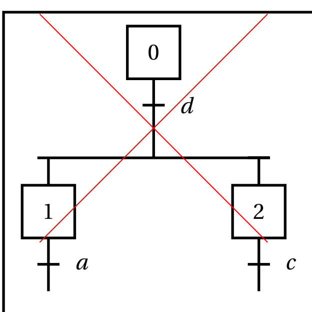
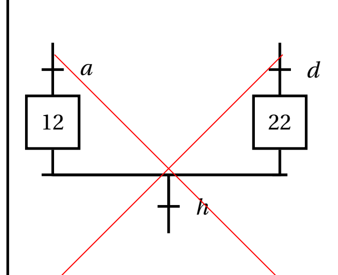
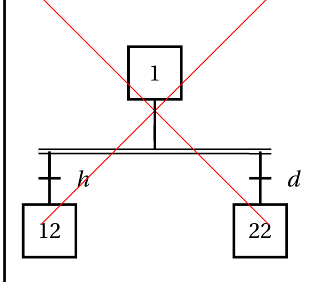
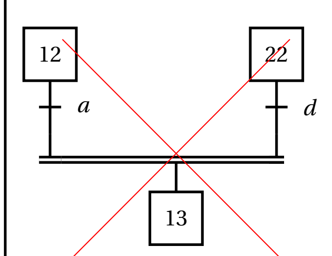
 La divergence en Ou ne doit pas être précédée par une transition ou suivie par des étapes	 La convergence en Ou ne doit pas être précédée par des étapes ou suivie par une transition.
 La divergence en ET ne doit pas être précédée par une étape ou suivie par des transitions,	 La convergence en ET ne doit pas être précédée par des transitions ou suivie par une étape.

Tableau 2.4: Règles de syntaxe

2.7 Règles d'évolution d'un GRAFCET

La modification de l'état de l'automatisme est appelée évolution, et est régie par 5 règles :

RÈGLE 1 : Situation initiale du GRAFCET

La situation initiale précise quelles étapes doivent être activées lors de la mise en service du système de commande de la machine. Cette situation est choisie par le concepteur. Elle est caractérisée par l'ensemble des **étapes initiales**.

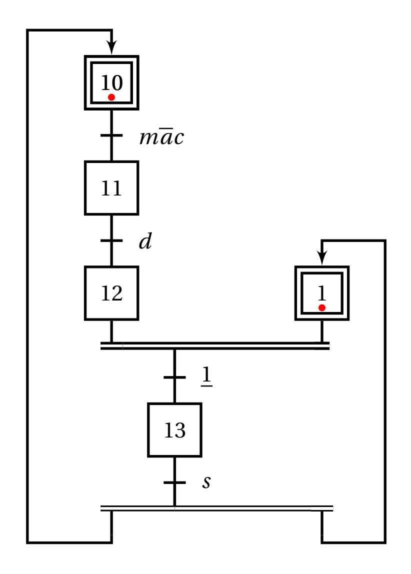
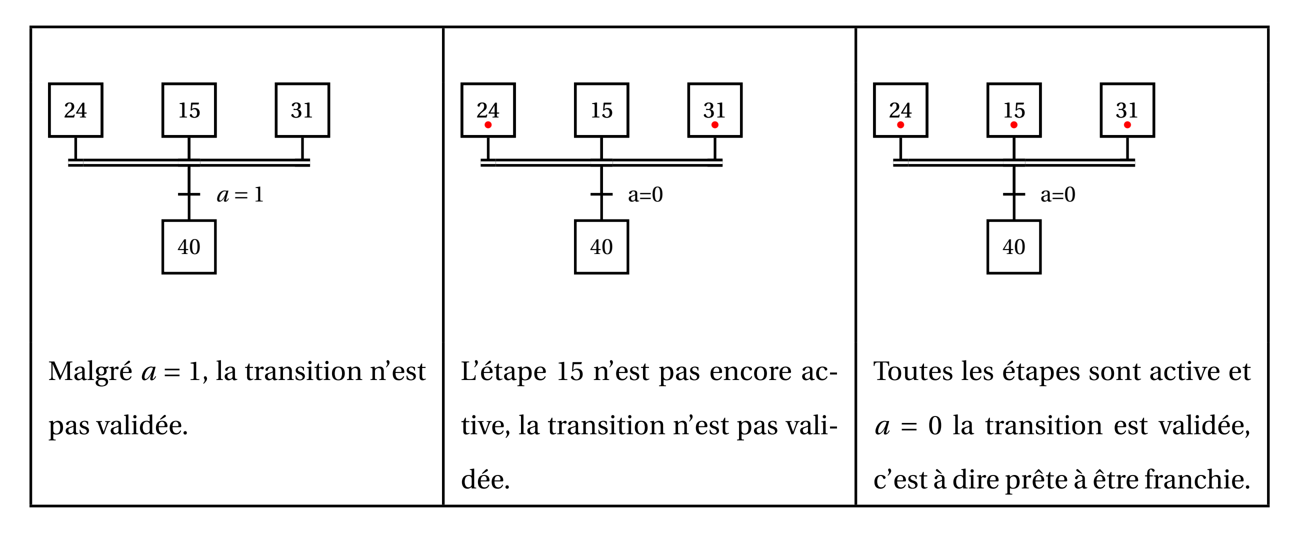


Figure 2.13: Règle 1

RÈGLE 2 : Transition franchissable

Une transition est franchissable et obligatoirement franchie si les deux conditions suivantes sont remplies :

- La transition est validée (toutes les étapes qui précèdent immédiatement la transition sont actives) ;
- La réceptivité associée à cette transition est vraie.

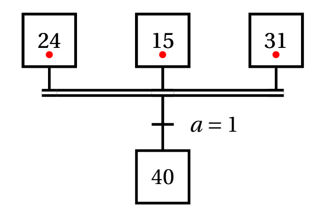


Malgré $a = 1$, la transition n'est pas validée.	L'étape 15 n'est pas encore active, la transition n'est pas validée.	Toutes les étapes sont active et $a = 0$ la transition est validée, c'est à dire prête à être franchie.

Tableau 2.5: Validation de transition

RÈGLE 3 : Franchissement d'une transition

Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

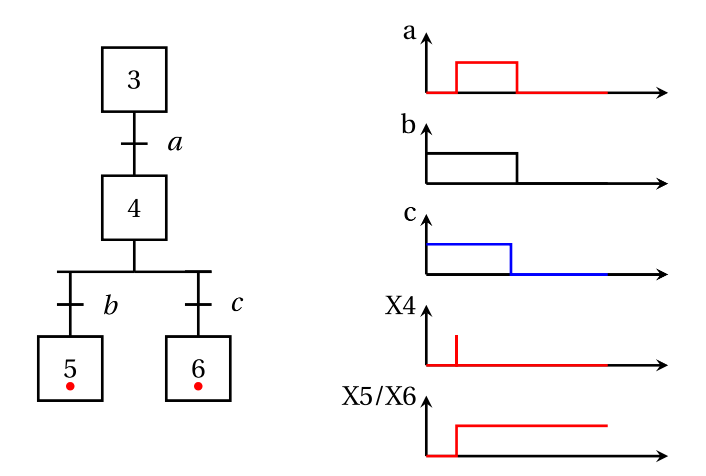


Les étapes sont actives et $a = 1$, la transition est obligatoirement franchie, c'est à dire désactivation des étapes 24, 15 et 31 et activation de l'étape 40.

Figure 2.14: Transition franchie

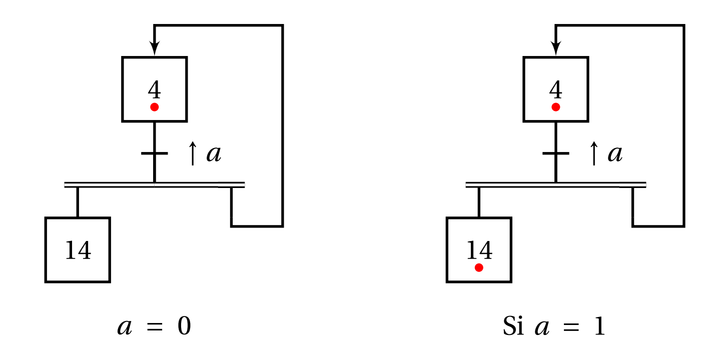
RÈGLE 4 : Franchissements simultanés

Plusieurs transitions simultanément franchissables sont simultanément franchies.



RÈGLE 5 : Activation et désactivation simultanées

Si au cours du fonctionnement, la même étape est simultanément activée et désactivée, elle reste active.



2.8 Mise en équation d'un GRAFCET

Mise en équation d'un GRAFCET

On peut matérialiser un GRAFCET en utilisant un automate programmable industriel (API),à l'aide de séquenceurs, de circuits logiques, circuits électriques etc. Par conséquent, le GRAFCET-doit être traduit par des équations logiques décrivant les conditions d'activation et de désactivation de chaque étape. Il existe plusieurs méthodes qui permettent d'obtenir les équations du GRAFCET. Les plus connues sont :

- la méthode des conditions du franchissement des transitions à ***évolution synchrone*** : méthode structurée et normalisée qui traduit directement les règles du GRAFCET
- la méthode "industrielle" à ***évolution asynchrone*** : cette méthode, bien que plus rapide à programmer, peut poser de problème pour quelques configuration du GRAFCET.

Ces méthodes passent par trois étapes :

1. la recherche des **fonctions transitions** : des équations des conditions de franchissement des transitions du GRAFCET,
2. l'écriture des **équations des étapes** du GRAFCET : l'étape sera caractérisée par une mémoire (bascule RS), dont il faut préciser l' entrée (S) pour l'activation et l' entrée (R) pour la désactivation,
3. l'établissement des **équations des actions**.

2.8.1 Fonctions transitions

Fonctions transitions

Considérons la séquence ci-contre d'un GRAFCET. D'après la 2eme règle d'évolution, on peut affirmer que :

- La transition $T_{n-1,n}$ peut être **franchie**, si elle est **validée** et la réceptivité associée est **vraie**,
- La transition $T_{n,n+1}$ peut être **franchie**, si elle est aussi **validée** et la réceptivité associée est **vraie**.

On définit alors deux conditions logiques dites «***fonctions transitions***» traduisant les affirmations précédentes :

$$Y_{n-1,n} = X_{n-1}.a$$

$$Y_{n,n+1} = X_{n}.b$$

2.8.2 Équations des étapes

Équations des étapes

Une mémoire (SR) est caractérisée par l'equation

$$X = S+\overline{R}X$$

L'équation de l'étape n peut être établie selon deux évolutions possibles :

Evolution synchrone : D'après la 3eme règle d'évolution, on a :

- l'étape n **s'active** si la transition $T_{n-1,n}$ est **franchie**
- l'étape n **se désactive** si la transition $T_{n,n+1}$ est **franchie**

Par conséquent, les entrées S et R de la mémoire seront

$$S = Y_{n-1,n}$$
$$R = Y_{n,n+1}$$

L'équation X_n s'écrit alors

$$X_n = Y_{n-1,n}+\overline{Y}_{n,n+1}X_n$$

L'évolution est dite **synchrone** car le franchissement de la transition $Y_{n,n+1}$ entraîne simultanément l'activation de l'étape X_{n+1} et la désactivation de l'étape X_n

Evolution asynchrone : Dans ce cas, la 3eme règle n'est pas vérifiée. les conditions d'activation et de désactivation sont comme suit :

- l'étape n **s'active** si la transition $T_{n-1,n}$ est **franchie**
- l'étape n **se désactive** l'étape suivante X_{n+1} est **active**

Par conséquent, les entrées S et R de la mémoire seront

$$S = Y_{n-1,n}$$
$$R = X_{n+1}$$

L'équation X_n s'écrit alors

$$X_n = Y_{n-1,n}+\overline{X}_{n+1}X_n$$

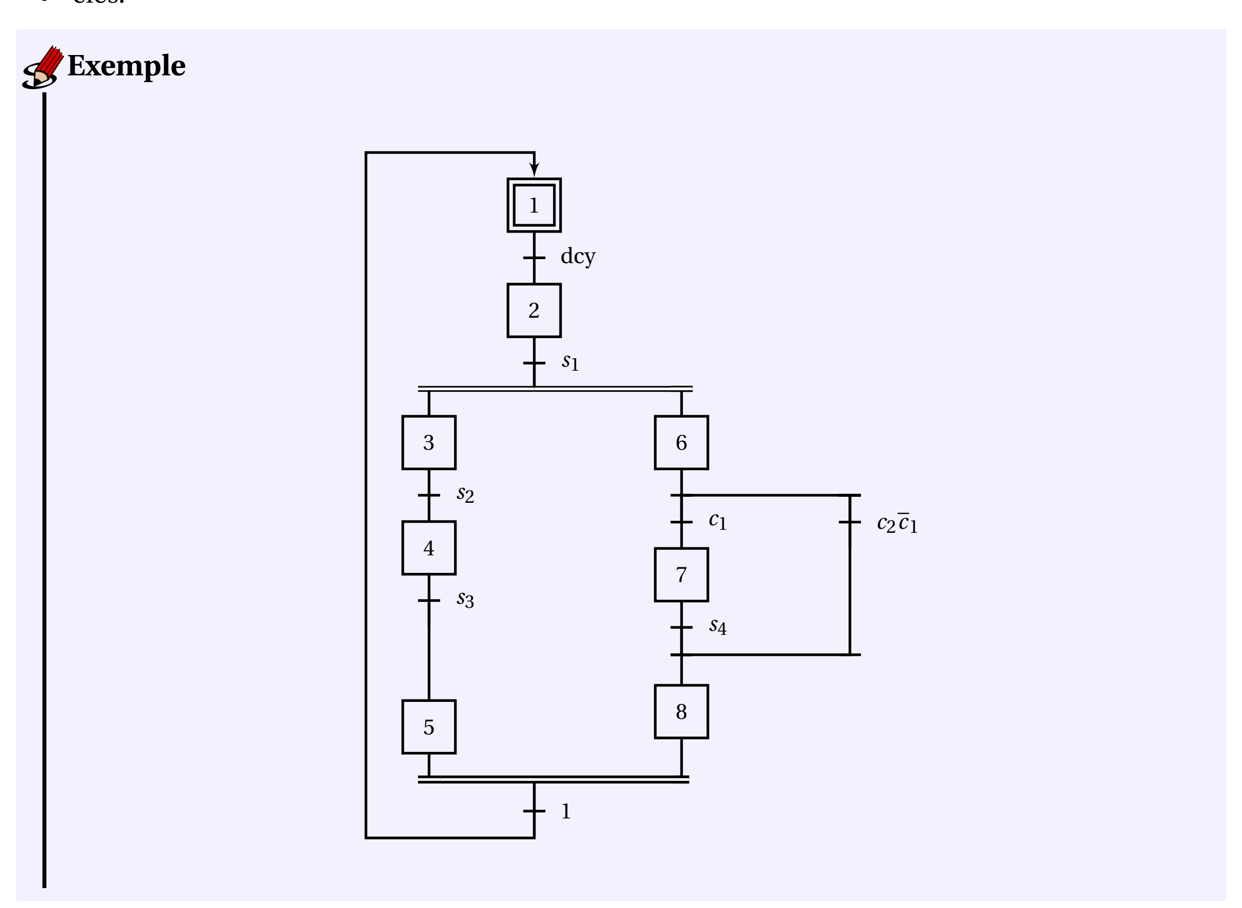
L'évolution est dite **asynchrone** car l'activation de l'étape X_{n+1} entraîne la désactivation de l'étape X_n

2.8.3 Équations des actions

Équations des actions

L'équation d'une action est la somme logique des variables X_j des étapes qui sont lui associés.

Exemple



Fonctions Transitions :

$$Y_{12} = dcyX_1$$

$$Y_{236} = s_1X_2$$

$$Y_{34} = s_2X_3$$

$$Y_{45} = s_3X_4$$

$$Y_{67} = c_1X_6$$

$$Y_{68} = c_2\bar{c}_1X_6$$

$$Y_{78} = s_4X_7$$

$$Y_{581} = X_5X_8$$

Équations des étapes en évolution synchrone :

Étape	Entrées S et R	Équations en évolution synchrone
1	$S = Y_{581}$ $R = Y_{12}$	$X_1 = Y_{581} + X_1\overline{Y}_{12} + In$
2	$S = Y_{12}$ $R = Y_{236}$	$X_2 = Y_{12} + X_2\overline{Y}_{236}$
3	$S = Y_{236}$ $R = Y_{34}$	$X_3 = Y_{236} + X_3\overline{Y}_{34}$
4	$S = Y_{34}$ $R = Y_{45}$	$X_4 = Y_{34} + X_4\overline{Y}_{45}$
5	$S = Y_{45}$ $R = Y_{581}$	$X_5 = Y_{45} + X_5\overline{Y}_{581}$
6	$S = Y_{236}$ $R = Y_{67} + Y_{68}$	$X_6 = Y_{236} + X_6\overline{Y}_{67}.\overline{Y}_{68}$
7	$S = Y_{67}$ $R = Y_{78}$	$X_7 = Y_{67} + X_7\overline{Y}_{78}$
8	$S = Y_{68} + Y_{78}$ $R = Y_{581}$	$X_8 = Y_{68} + Y_{78} + X_8\overline{Y}_{581}$

Équations des étapes en évolution asynchrone :

Étape	Entrées S et R	Équations en évolution asynchrone
1	$S = Y_{581}$ $R = X_2$	$X_1 = Y_{581}+X_1\overline{X}_2+In$
2	$S = Y_{12}$ $R = X_3.X_6$	$X_2 = Y_{12}+X_2(\overline{X}_3+\overline{X}_6)$
3	$S = Y_{236}$ $R = X_4$	$X_3 = Y_{236}+X_3\overline{X}_4$
4	$S = Y_{34}$ $R = X_5$	$X_4 = Y_{34}+X_4\overline{X}_5$
5	$S = Y_{45}$ $R = X_1$	$X_5 = Y_{45}+X_5\overline{X}_1$
6	$S = Y_{236}$ $R = X_7+X_8$	$X_6 = Y_{236}+X_6\overline{X}_7.\overline{X}_8$
7	$S = Y_{67}$ $R = X_8$	$X_7 = Y_{67}+X_7\overline{X}_8$
8	$S = Y_{68}+Y_{68}$ $R = X_1$	$X_8 = Y_{68}+Y_{78}+X_8\overline{X}_1$

2.9 Extension du modèle GRAFCET

2.9.1 Étapes sources-Étapes puits

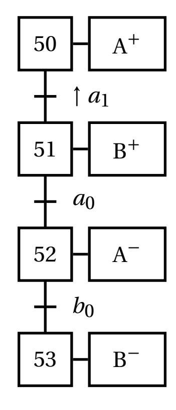


Figure 2.15: Étape source-Étape puits

Étapes sources

On appelle étape source une étape non reliée à une transition amont Figure (2.15). Cette étape ne peut être activée que par un ordre de forçage. Une étape initiale sans transition amont est une étape source activée inconditionnellement à la mise sous tension.

Pour montrer la mise en équation d'une étape source, supposons que l'étape 50 est forcée par l'étape 12 d'un GRAFCET maître, alors l'équation sera :

$$Y_{5051} = X_{50}\, a_1$$
$$X_{50} = X_{12} + X_{50}\overline{Y}_{5051}$$

Étapes puits

On appelle étape puits une étape non suivie d'une transition Figure (2.15). Cette étape ne peut être désactivée que par un ordre de forçage.

Supposons que l'étape 53 soit désactivée par l'étape 9 d'un GRAFCET maître , alors l'équation sera :

$$Y_{5253} = X_{52}\, b_0$$
$$X_{53} = Y_{5253} + X_{53}\overline{X}_9$$

2.9.2 Transitions sources-Transitions puits

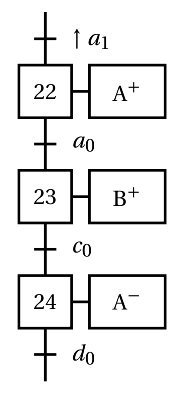


Figure 2.16: Transition source-Transition puit

Transitions sources

Une transition source est une transition non précédée d'une étape Figure (2.16). Par convention cette transition est toujours validée (malgré l'absence d'étape précédente), pour la franchir il suffit que la réceptivité soit vraie. La réceptivité associée à une transition source est en générale une réceptivité sur front (sinon l'étape suivante est activée en permanence).

Ainsi, l'équation de l'étape 22 serait :

$$Y_{2223} = X_{22} a_0$$
$$X_{22} = \uparrow a_1 + X_{22} \overline{Y}_{2223}$$

Transitions puits

Une transition puits est une transition non suivie d'une étape, Figure (2.16). Les règles de franchissement s'appliquent à ce type de transition, lors du franchissement l'étape précédente est désactivée, aucune autre étape n'est activée.

Ainsi l'équation de l'étape 24 sera :

$$Y_{2324} = X_{23} c_0$$
$$Y_{24} = X_{24} d_0$$
$$X_{24} = Y_{2324} + X_{24} \overline{Y}_{24}$$

Chapitre 3

Structuration d'un GRAFCET

3.1 Introduction

Les Systèmes Automatisés de production sont de plus en plus complexes, afin de simplifier l'étude, la mise en oeuvre et la maintenance du système, il est nécessaire de structurer la partie commande et la partie opérative. L'objectif essentiel de la hiérarchisation est de permettre une approche progressive structurée du fonctionnement d'un système automatisé, tant au niveau de l'analyse qu'au niveau de la représentation.

L'analyse structurée d'un système permet de décrire celui-ci depuis le niveau le plus général vers des niveaux de plus en plus détaillé. Cette structuration utilise les notions de tâches et de Macro-étape. Le système est décomposé soit suivant sa topologie (les tâches correspondant à des parties opératives indépendantes) soit fonctionnellement (les tâches correspondant à des fonctions du système). Les commandes de forçage et figeage de GRAFCET, sont des moyens supplémentaires qui permettent de préciser la hiérarchie des différents GRAFCETS.

La hiérarchie des différents GRAFCETS issue de cette structuration est celle qui vient de la description du fonctionnement (succession séquentielle des tâches définie par le processus). En fait il est souvent nécessaire de placer à un niveau hiérarchiquement supérieur des GRAFCETs de gestion. Les principaux GRAFCETs que l'on peut trouver sont :

GRAFCET de surveillance : (de sécurité) Ce GRAFCET décrit l'ensemble de procédures de sécurité du système, c'est le GRAFCET hiérarchiquement le plus important. L'arrêt d'urgence et les procédures de mise en route sont décrits dans ce GRAFCET.

GRAFCET de conduite : (ou GRAFCET des Modes de Marches) Ce GRAFCET décrit l'ensemble de procédures de marches (automatique, Cycle/Cycle, Manuel,...) et des arrêts normaux.

GRAFCET de maintenance : Ensembles de procédures de réglage de la partie opérative.

GRAFCET de Production : Ce GRAFCET est le niveau de description du fonctionnement normal de l'automatisme. Ce GRAFCET est en général décomposé en plusieurs tâches représentant les différentes fonctions de l'automatisme.

3.2 Macro-étape

3.2.1 Principe

Le concept de macro-étape est à différencier de celui de sous-programme, une macro-étape est plutôt une extension du concept d'étape . Les macro-étapes sont à utiliser pour faciliter la représentation de systèmes complexes dans un but de structuration. Le concept de macro-étape se prête aisément à l'analyse descendante du système. A chaque Macro-étape correspond une expansion et réciproquement, si plusieurs cycles identiques sont à représenter, autant de macro et d'extension seront nécessaires.

3.2.2 Symbole

Macro-étape

Une macro-étape (ME) est la représentation unique d'un ensemble d'étapes et de transition nommé «***Expansion d'étapes***», la macro-étape se substitue à une étape du GRAFCET, Figure (3.1).

1. L'expansion de ME comporte une étape d'entrée et une étape de sortie repérées par E et S.
2. Tout franchissement de la transition amont de la macro-étape active l'étape E d'entrée de son Expansion.
3. L'étape de sortie participe à la validation des transition aval de la macro-étape.
4. La transition suivant la macro-étape n'est validée que lorsque la dernière étape de l'expansion de macro-étape est active.

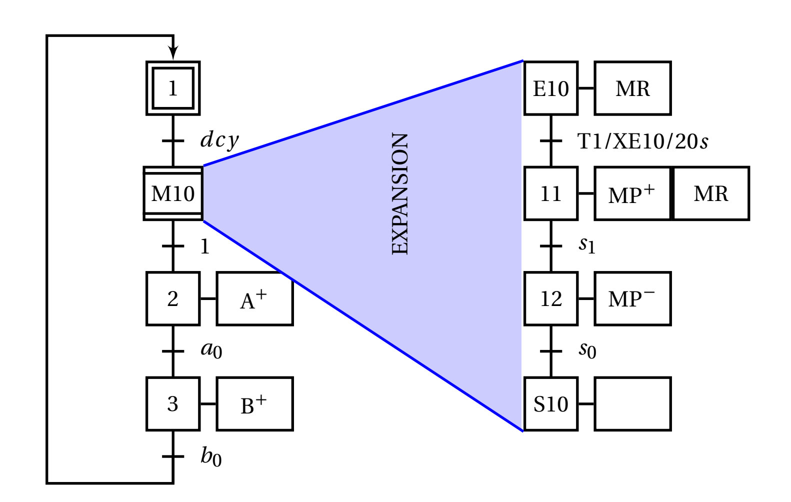


Figure 3.1: Macro-étape et expansion

Lorsque la transition T_{1-M10} est franchie, alors la macro-étape M10 est activée, ce qui entraîne l'activation de l'étape E10 de l'entrée de l'expansion et le cycle décrit dans l'expansion se déroule jusqu'à l'étape de sortie S10. Dès que l'étape S10 est active, la transition T_{M10-2} sera franchie. Il est préférable de ne pas associer d'actions aux étapes d'entrée et de sortie de la macro-étape.
L'equation de M10 en évolution synchrone est

$$X_{M10} = Y_{1-M10} + \overline{Y}_{M10-2} X_{M10}$$

avec

$$Y_{1-M10} = X_1 dcy, \qquad Y_{M102} = X_{S10}.X_{M10}$$

Exemple

On donne le GRAFCET suivant, rétablir un GRAFCET equivalent à base de macro-étapes.

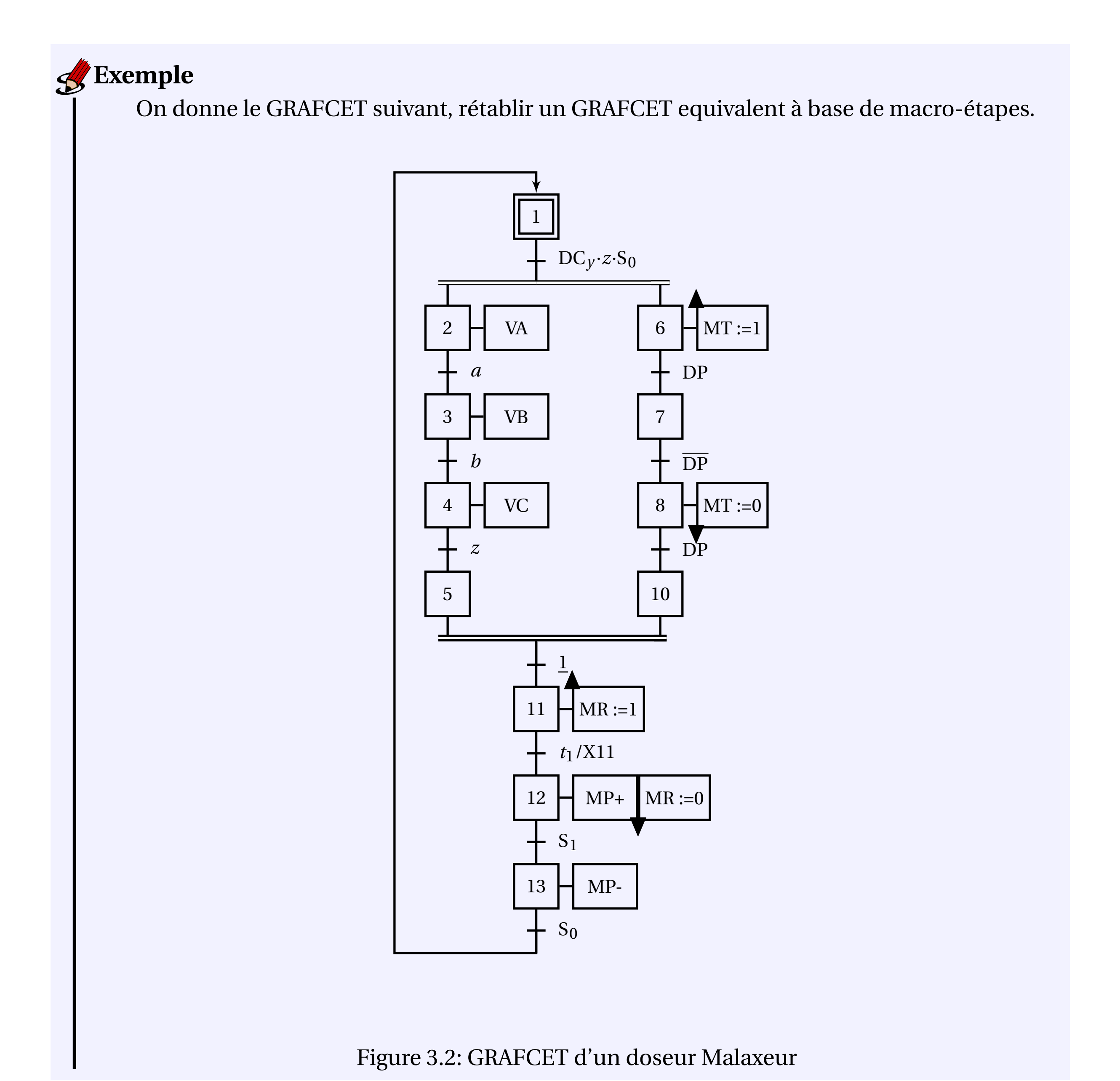


Figure 3.2: GRAFCET d'un doseur Malaxeur

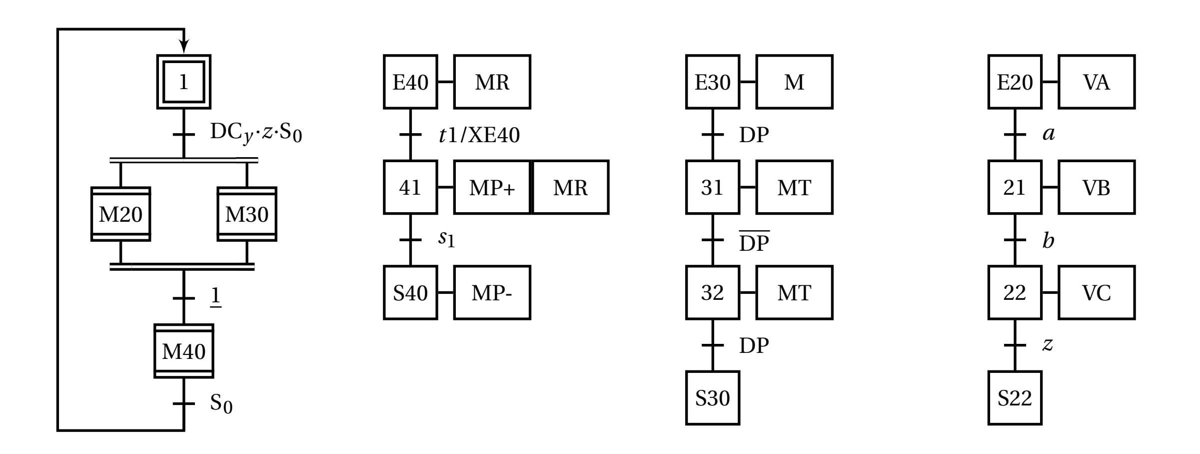


Figure 3.3: Solution avec des macro-étapes

3.3 Tâches, sous-programmes

Tâches, sous-programmes

Le concept de tâche permet lui aussi de décrire les systèmes de façon structurée, il correspond à la notion de sous programme, Figure (3.4).

Lorsque une tâche doit être réalisée plusieurs fois dans un cycle, la programmation sous forme de tâche permet de simplifier l'analyse et la programmation.

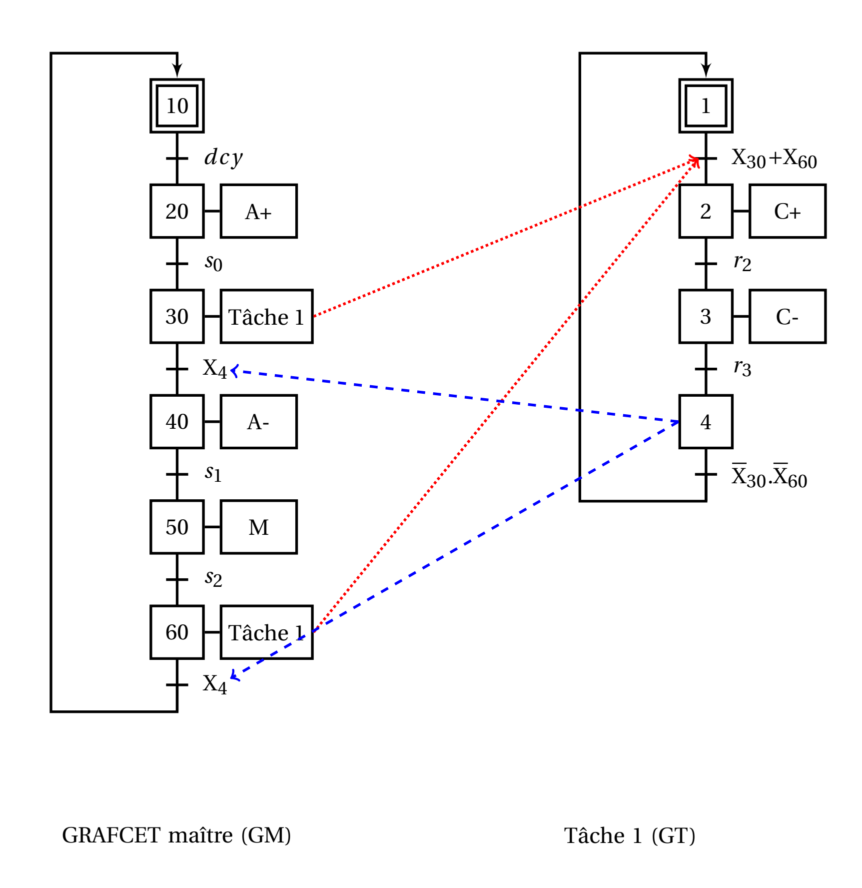


Figure 3.4: Tâche-Sous programme

Supposons que les GRAFCETs (GM) et (GT) sont initialisés.

- Si l'étape 30 est active, la transition T_{12} sera franchie et le GRAFCET (GT) s'exécute,
- Si l'étape 4 (étape vide) est active, la transition T_{3040} sera franchie et le GRAFCET (GM) continue son cycle
- Si l'étape 40 est active ($\overline{X}_{30}.\overline{X}_{60} = 1$), la transition T_{41} sera franchie et la GRAFCET (GT) s'initialise pour un nouveau cycle.



Exemple

On reprend le GRAFCET de la Figure (3.3) où la dernière séquence sera remplacée par une tâche.

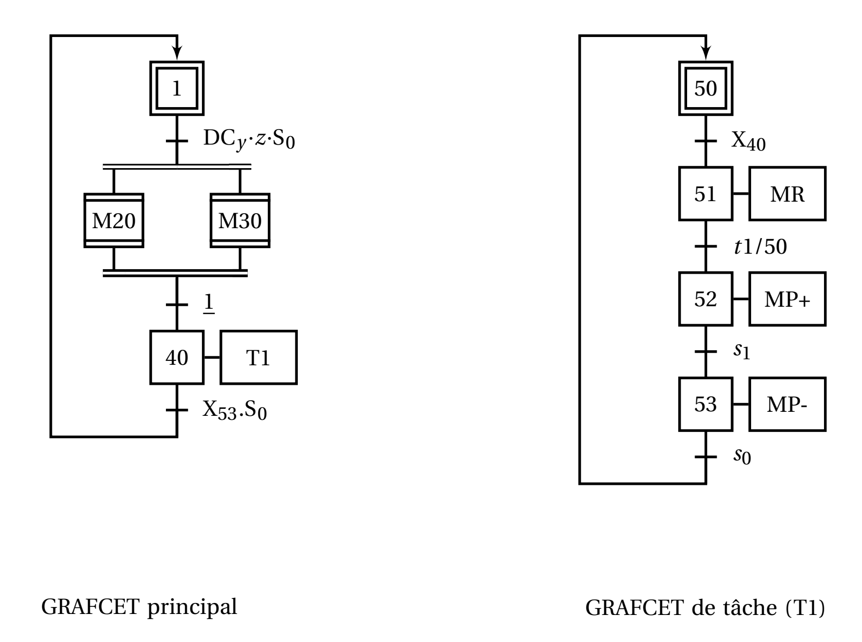


Figure 3.5: Solution avec des macro-étapes et tâche

3.4 Forçage et Figeage

Le forçage n'est pas a proprement parlé une méthode de structuration des GRAFCETs, mais c'est un outil nécessaire à la hiérarchisation des GRAFCETs. La description du fonctionnement normal est facilitée par les outils d'analyse descendante, par contre les fonctionnements anormaux (sécurité, arrêt d'urgence) qui nécessite une prise de décision et une action rapide nécessite un outil capable d'inhiber le déroulement normal du GRAFCET, pour autoriser la réalisation de cycle de sécurité.

3.4.1 Définition

Forçage

- Par **forçage** de situation, on entend le passage imposé de la situation courante du GRAFCET désigné à une situation déterminée différente de celle qu'aurait atteint celui-ci s'il avait évolué normalement.
- Le GRAFCET forcé évolue alors *sans franchissement de transition*.
- L'ordre de **forçage** ne peut être émis que par un GRAFCET de *niveau hiérarchiquement supérieur* vers un GRAFCET *de niveau inférieur*, Figure (3.6). Ces ordres de forçage sont utiles principalement pour prendre en compte les sécurités.

3.4.2 Règles de forçage

Règles de forçage

Le forçage est un ordre interne au GRAFCET consécutif a une evolution. Pour une situation comportant plusieurs ordres de forçage, les GRAFCET forcés prendront immédiatement et directement la ou les situations imposées.

1. Toute apparition d'une situation de **forçage** est **prioritaire** par rapport a toute autre activité du GRAFCET.
2. Les GRAFCET forcés sont **maintenus** dans la situation de forçage tant que *l'ordre de forçage est valide.*
3. *Un GRAFCET ne peut être* ***forcé*** *que par un GRAFCET* hiérarchiquement supérieur.
4. A tout instant un GRAFCET ne peut être **forcé** que dans *une situation et une seule* à partir d'un et *un seul GRAFCET supérieur.*

3.4.3 Représentation

Représentation du forçage

De manière générale le forçage est représenté dans le GRAFCET par la séquence suivante :

F/nom du GRAFCET : (Situation)

Le forçage est inscrit dans un cadre d'action.

3.4.4 Types de forçage

Le forçage décrit comment doit se retrouver le(s) GRAFCET(s) forcé(s) après le forçage, on distingue plusieurs situation types :

Forçage dans une situation non vide

- Le forçage dans une situation non vide correspond à l'activation d'une ou plusieurs étapes dès que l'étape de forçage du GRAFCET supérieur est activée, Figure (3.6).
- Il est possible de forcer plusieurs étapes. Le forçage est effectué en dehors des règles du GRAFCET.
- Ce forçage est principalement utilisé dans des procédures d'initialisation mais aussi pour activer un cycle particulier après un arrêt d'urgence.

Remarque : Pour que la situation évolue, il faut que le forçage cesse, sinon le GRAFCETforcé reste dans la situation de forçage.

Forçage dans une situation vide

- Le **forçage** dans une *situation vide* correspond à la **désactivation** de toutes les étapes d'un GRAFCET dès que l'étape de forçage du GRAFCET supérieur est activée, Figure (3.6).

– Le *forçage* dans une *situation vide* nécessite par la suite le *forçage* dans une **situation non vide** pour pouvoir **redémarrer**.

Remarque : Attention de ne pas forcer dans une situation vide tous les GRAFCETs du système (impossible de redémarrer).

Exemple

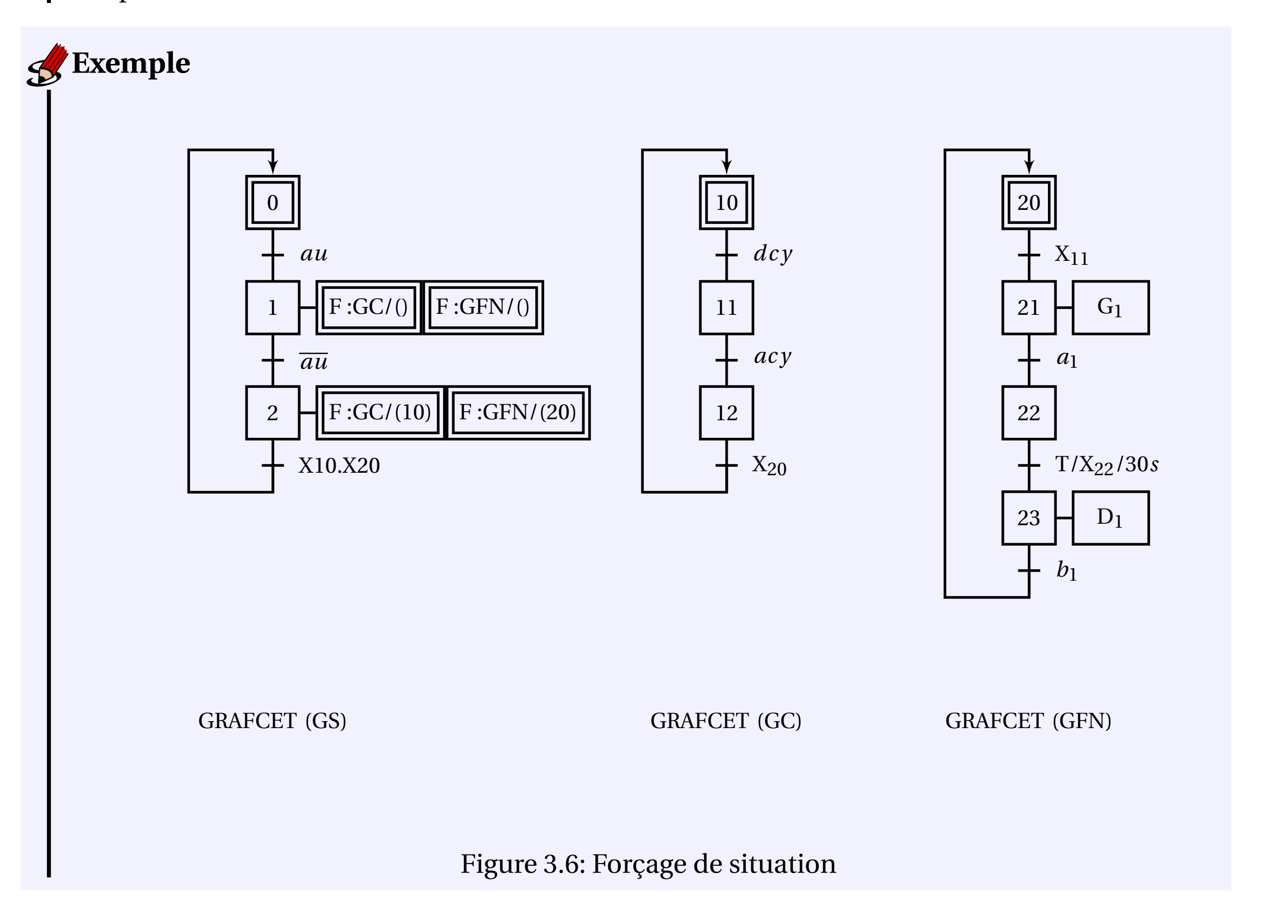


Figure 3.6: Forçage de situation

Supposons que les étapes 0, 11 et 21 sont actives.

– Si on appuie sur le bouton d'arrêt d'urgence au ($au = 1$), il y aura

1. l'étape 1 s'active et un ordre de forçage apparaît,
2. il s'agit d'un forçage des GRAFCETs (GC) et (GFN) dans une situation vide. Par conséquent toutes les étapes des GRAFCETs (GC) et (GFN) se désactivent,

– Si on relâche le bouton d'arrêt d'urgence au ($au = 0$),

1. l'étape 2 s'active et un ordre de forçage apparaît,
2. il s'agit d'un forçage des GRAFCETs (GC) et (GFN) dans les étapes 10 et 20, respectivement. Par conséquent ces étapes s'activent, la transition T_{20} sera franchie et on reprend la situation de fonction normal quand l'étape 0 sera active.

Figeage

– Le figeage est un cas particulier du forçage dans lequel l'évolution normale du GRAFCET est figée, Le GRAFCET figée voit sont évolution stoppée dès l'apparition de l'ordre de figeage.

- le figeage est représenté dans le GRAFCET par la séquence suivante :

F/nom du GRAFCET : (* situation)

- Le figeage est à utiliser avec précaution, en effet les ordres émis par les étapes restent vrais indépendamment de l'évolution des réceptivités.
- Il est souvent nécessaire de bloquer simultanément les sorties.

Chapitre 4

Le GEMMA

Le mot GEMMA est l'acronyme de **G**uide d'**É**tude des **M**odes de **M**arches et d'**A**rrêts Le GEMMA est un guide d'étude qui permet de structurer la partie commande d'un système automatisé de production.

Le GEMMA permet de décrire le fonctionnement du système dans tous les modes de marche, la partie opérative étant sous contrôle de la partie commande. Le GEMMA n'intervient donc que si la partie commande est en énergie et active.

Le GEMMA définit donc, à tout instant, l'état dans lequel se trouve la partie commande du système automatisé de production.

Le GEMMA se compose d'une grille à remplir. Cette grille est constituée de rectangles d'état appelés modes. Ces rectangles sont reliés entre eux par des liaisons orientées. Le passage d'un rectangle à l'autre s'effectue un peu à la manière du franchissement d'une transition de GRAFCET.

Le GEMMA n'est pas un outil figé, il est modifiable à volonté en fonction des spécifications à obtenir. Les liaisons orientées présentes sur le document de référence ne sont là qu'à titre indicatif.

4.1 Concepts de bases du GEMMA

4.1.1 Système en ordre de marche

Le guide graphique GEMMA, est constitué de deux zones

– **Partie commande hors énergie** : Dans cet état la partie opérative n'est pas sous le contrôle de la partie commande. La partie opérative peut être en énergie ou hors énergie. La sécurité est garantie par les choix technologiques et la procédure de mise en énergie de la partie opérative. Cette zone du GEMMA, située à l'extrême gauche, correspond à l'état inopérant de la partie commande.

– **Partie commande sous énergie** : C'est la partie qui va nous permettre de définir les différents modes de marche et d'arrêt de notre machine ainsi que les conditions de passage d'un mode à l'autre. Cette partie est subdivisée en trois zones ou en trois familles de procédures.

4.1.2 Marche en production

On dira que le système est en « production » si la valeur ajoutée pour laquelle le système a été conçu est obtenue. On dira que le système est « hors production » dans le cas contraire . La zone de production se situe dans la zone partie commande (PC) sous énergie.

4.1.3 Familles de modes de marches et d'arrêts

On peut classer en trois grandes familles les modes de marches et d'arrêts d'un système automatisé fig (4.1) :

Les procédures d'Arrêt zone A : Une marche automatique fonctionne rarement 24 heures sur 24. Il est nécessaire de l'arrêter de temps à autre, pour des raisons extérieures au système (la journée est finie) ou bien par manque d'approvisionnement. On classera dans cette famille tous les modes conduisant à (ou traduisant) un état d'arrêt du système pour des raisons extérieures. Ils sont regroupés dans une zone A, « procédures d'Arrêt », du guide graphique.

Les procédures de Fonctionnement zone F : On groupe dans cette famille tous les modes ou états qui sont indispensables à l'obtention de la valeur ajoutée, ou, autrement dit, tous ceux sans lesquels on ne peut pas techniquement ou fonctionnellement obtenir la valeur ajoutée pour laquelle la machine est prévue. Ces modes sont regroupés dans le guide graphique dans une zone F, « procédures de Fonctionnement ». Les modes préparatoires à la production, de réglages ou, de tests, peuvent faire partie de cette famille.

Les procédures de Défaillance zone D : Il est rare qu'un système fonctionne sans incident pendant toute sa vie : il est indispensable de prévoir les défaillances. On regroupera dans cette famille tous les modes conduisant à (ou traduisant) un état d'arrêt du système pour des raisons intérieures au système, autrement dit, à cause de défaillances de la partie opérative. Ces modes sont regroupés dans une zone D, « procédures de Défaillance », du guide graphique.

4.2 Les procédures d'arrêt ou les états (A)

Les états (A)

A1-Arrêt dans l'état initial C'est l'état « repos » de la machine. C'est l'état dans lequel se trouve la machine avant de passer en production normale. Ce rectangle est repéré sur le GEMMA par un double encadrement. Pour une étude plus facile de l'automatisme, il est recommandé de représenter la machine dans son état initial.

A2-Arrêt demandé en fin de cycle Lorsque l'arrêt est demandé, la machine continue de produire jusqu'à la fin du cycle. A2 est donc un état transitoire vers l'état A1.

A3-Arrêt demandé dans un état déterminé La machine continue de produire jusqu'à un arrêt en une position autre que la fin de cycle. C'est un état transitoire vers A4.

A4-Arrêt obtenu La machine est alors arrêtée en une autre position que la fin de cycle.

A5-Préparation pour remise en route après défaillance C'est dans cet état que l'on procède à toutes les opérations (dégagement, nettoyage, ...) nécessaires à une remise en route après défaillance.

A6-Mise de la PO dans l'état initial La machine étant en A6, on remet manuellement et/ou automatiquement la partie opérative en position pour un redémarrage dans l'état initial.

A7-Mise de la PO dans un état déterminé La machine étant en A7, on remet la partie opérative en position pour un redémarrage dans une position autre que l'état initial.

4.3 Les procédures de fonctionnement ou les états (F)

les états (F)

F1-Production normale Dans cet état, la machine produit normalement. C'est l'état pour lequel elle a été conçue, la valeur ajoutée produite correspond au cahier des charges client. Ce rectangle est repéré sur le GEMMA par un encadrement renforcé. On parle souvent de mode automatique.

F2-Marche de préparation C'est un état utilisé pour des machines nécessitant une préparation (préchauffage d'un outillage, remplissage de cuves, ...) préalable à la production normale. C'est un état transitoire vers l'état F1.

F3-Marche de clôture A l'inverse cet état est utilisé pour des machines nécessitant différentes opérations (nettoyages, vidanges, ...) en fin de journée ou en fin de série.

F4-Marche de vérification dans le désordre Cet état permet de vérifier, de régler certaines fonctions, certains mouvements sur la machine sans respecter l'ordre du cycle de production normale. On parle souvent de mode manuel.

F5-Marche de vérification dans l'ordre Dans cet état, le cycle de production peut être exploré au rythme voulu (pas à pas, cycle par cycle, ...) par la personne effectuant la vérification. Dans cet état la machine peut produire ou ne pas produire.

F6-Marche de test Cet état permet des opérations de réglage la machine étant en production ou hors production. Ces opérations de réglage doivent permettre à la machine de produire la valeur ajoutée attendue.

4.4 Les procédures de défaillances ou états (D)

Les états (D)

D1-Arrêt d'urgence C'est l'état pris lors d'un arrêt d'urgence. On y prévoit non seulement les arrêts mais aussi les cycles de dégagement, les procédures et précautions nécessaires pour éviter ou limiter les conséquences dues à la défaillance.

D2-Diagnostic et/ou traitement de défaillance C'est dans cet état que la machine peut être examinée après une défaillance et qu'il peut être apporté un traitement permettant le redémarrage.

D3-Production tout de même Cet état permet de produire après une défaillance. On parle de « production dégradée » ou de « production forcée ». Ce peut être par exemple une production aidée par des opérateurs non prévus en production normale.

4.5 Mode d'emploi du GEMMA

Avec le GEMMA, l'étude des modes de marches et d'arrêts est prévue dés la conception de la machine et intégrée dans sa réalisation. Sa mise en oeuvre aura lieu après que les spécifications fonctionnelles ont été analysées par les GRAFCETs. Ces spécifications fonctionnelles ne concernent que le fonctionnement normal. Le GEMMA s'attarde sur les spécifications opérationnelles qui envisage les différents modes de marches et d'arrêts, ainsi que les divers cas de défaillance.

Chaque rectangle d'état est défini par son appartenance à l'une des trois familles de procédures, son appartenance ou non à la zone de production, son repère et sa désignation. L'automaticien peut compléter la désignation de chaque rectangle d'état.

4.5.1 Sélection des modes de marches et d'arrêts

Dans un premier temps l'automaticien va choisir les rectangles d'état nécessaires à la description du système automatisé de production. Pour chaque rectangle d'état choisi il donnera une description précise soit de l'état de la partie opérative soit des procédures correspondantes au fonctionnement attendu.

- si le mode proposé est retenu, il sera précisé en langage littéral de fonctionnement propre à la machine, dans le « rectangle-état » ;
- si le mode proposé n'est pas nécessaire pour la machine, une croix est portée dans le « rectangle état », pour bien signifier qu'il n'est pas retenu.

4.5.2 Conditions d'évolution entre modes de marches et d'arrêts

Dans un deuxième temps, l'automaticien précisera les conditions de passage d'un rectangle d'état à l'autre. Il établira ainsi plusieurs boucles opérationnelles. Une boucle opérationnelle est une succession d'états caractérisant le fonctionnement du système automatisé de production.
L'élaboration de ces conditions de passage rend possible la conception du pupitre de commande et entraîne éventuellement l'adjonction de capteurs supplémentaires. Le passage d'un état vers l'autre s'effectue de deux façons :

- soit avec une condition d'évolution qui est portée sur la liaison orientée entre les deux états et concrétisée par un capteur sur machine ou par un auxiliaire de commande sur pupitre ;
- soit sans condition d'évolution (si cela n'apporte aucune information complémentaire).

4.6 Finalité du GEMMA

4.6.1 Réaliser le GRAFCET de CONDUITE (GC)

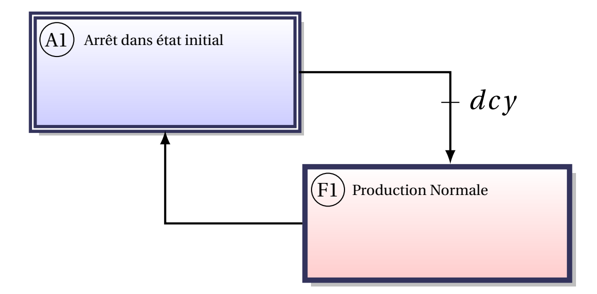
Le GRAFCET de CONDUITE (GC) doit permettre d'assurer la bonne marche du système automatisé en intégrant les dispositions précisées par le GEMMA lors de l'étude des modes de marches et d'arrêts.

4.6.2 Définir le pupitre de commande

La conception et l'organisation du pupitre de commande découle directement des conditions d'évolution des modes de marches et d'arrêts formalisés par le GEMMA. Le choix et la localisation des organes de dialogue (boutons, voyants, etc.) tiennent compte des spécifications de conduite et de sûreté, des règles de l'art (normes, réglementations) et des objectifs ergonomiques.

4.7 Boucles opérationnelles classiques

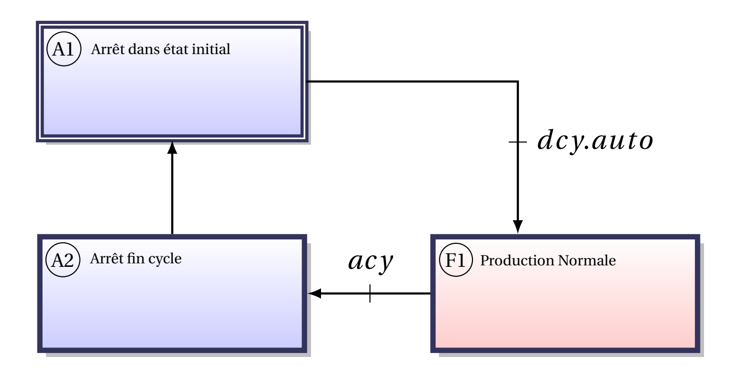
4.7.1 Marche de production cycle par cycle



Ce GEMMA correspond au cas ou la machine produit uniquement à la pièce. Ce mode de fonctionnement est appelé mode semi-automatique. Le GRAFCET résultant est directement le GRAFCET

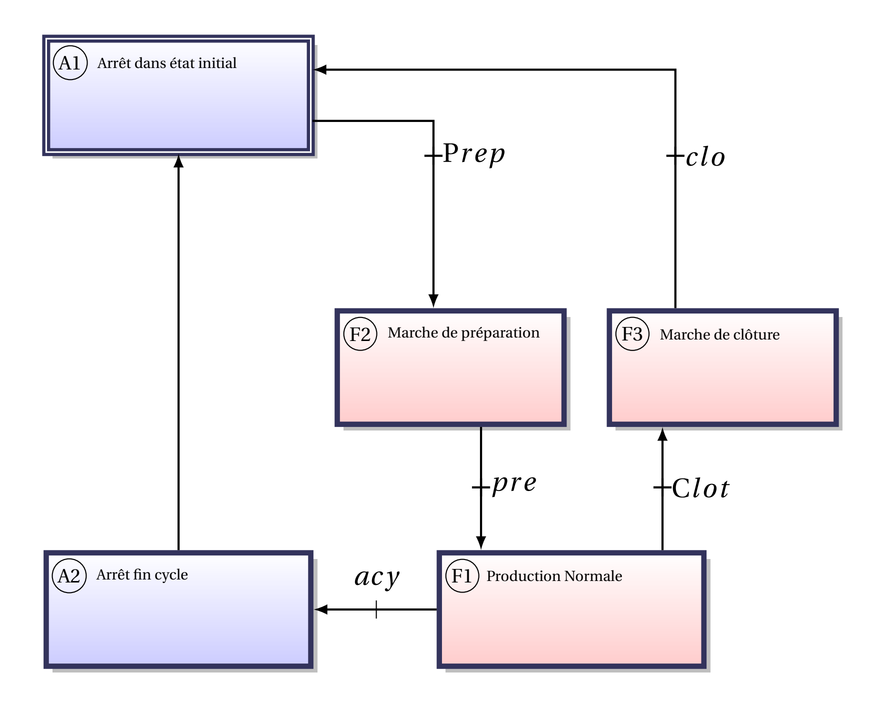
de base. La condition de mise en marche, c'est que la machine soit en condition initiale et que l'opérateur appui sur le bouton de départ de cycle.

4.7.2 Marche de production automatique



Ce GEMMA correspond au cas ou la machine produit en mode automatique. La condition de mise en marche est que le sélecteur de mode soit à automatique, que le bouton de départ de cycle soit appuyé et que la machine soit en condition initiale. Lorsque la machine produit, elle peut s'arrêter en fin de cycle suite à l'appui du bouton d'arrêt de cycle (acy).

4.7.3 Marche de production automatique avec marche de préparation et de clôture



Ce GEMMA correspond au cas ou la machine produit en mode automatique avec une marche de préparation et de clôture. Le rectangle-état F2 est utilisé lorsque l'automatisme exige certaines

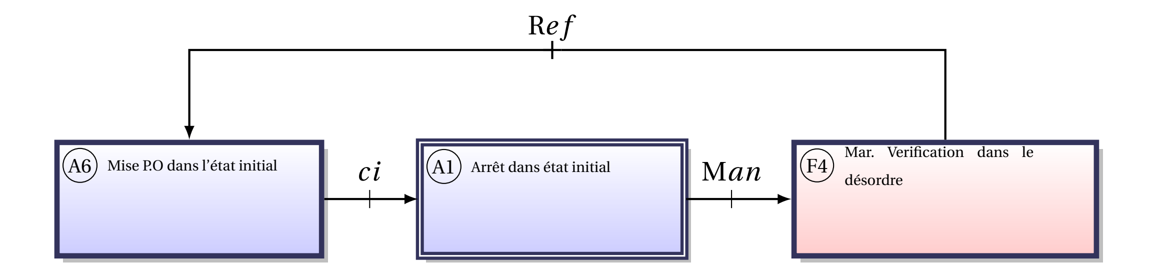
préparations avant de passer en production normale. Ces préparations peuvent être faites automatiquement ou manuellement. Pendant cette phase de préparation, il peut y avoir production ou non, ce qui explique que ce rectangle-état chevauche la frontière entre la zone «en production» et la zone «hors production».

Par exemple, si l'automatisme est une unité de perçage, il faut actionner le moteur de perçage avant d'entrer en production.

Lorsque la machine exige de faire certaines opérations (en plus de la fin de cycle normal) pour retourner en condition initiale, il faut prévoir une marche dite de clôture. Cette marche peut être faite automatiquement ou manuellement. Selon les actions à faire, il y a production ou non, ce qui explique que ce rectangle-état chevauche la frontière entre la zone «en production» et la zone «hors production».

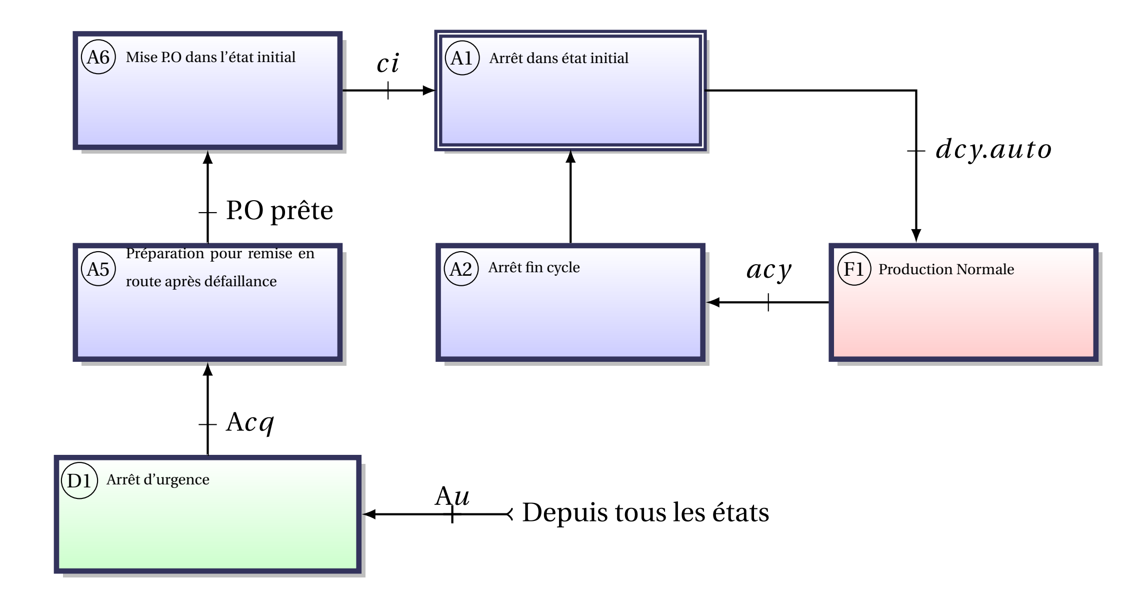
Par exemple, on peut prévoir une marche de clôture pour indiquer à l'opérateur de nettoyer la machine. On peut aussi y prévoir la vidange d'un convoyeur et l'arrêt progressif des postes d'une machine.

4.7.4 Marche de vérification dans le désordre et mise P.O. dans état initial



C'est la boucle qui permet d'atteindre le mode « manuel ». ce mode permet à l'opérateur de pouvoir tester les actionneurs, pré-actionneurs, capteurs, etc., du système, dans le désordre. A la fin d'une marche de vérification dans le désordre, la machine n'est pas dans son état initial, il y a donc obligatoirement passage par le rectangle d'état A6.

4.7.5 Marche automatique avec arrêt d'urgence



Dans le cas ou un arrêt d'urgence causé par une défaillance grave est envisagé, il faut pouvoir aller au rectangle-état D1 lorsque cet arrêt d'urgence se produit. Et cela, quelque soit le rectangle-état ou la machine se situe. Pour éviter d'encombrer le GEMMA, on met simplement en évidence l'évolution de F1 vers D1 et on ajoute un symbole de regroupement avec la mention «Depuis tous les états». Après l'arrêt d'urgence, il faut préparer la machine à sa remise en route en la nettoyant ou en dégageant les pièces coincées (F5), puis remettre la partie opérative en condition initiale de façon manuelle ou par une initialisation automatisée (F6). La machine sera alors prête à être redémarrée.

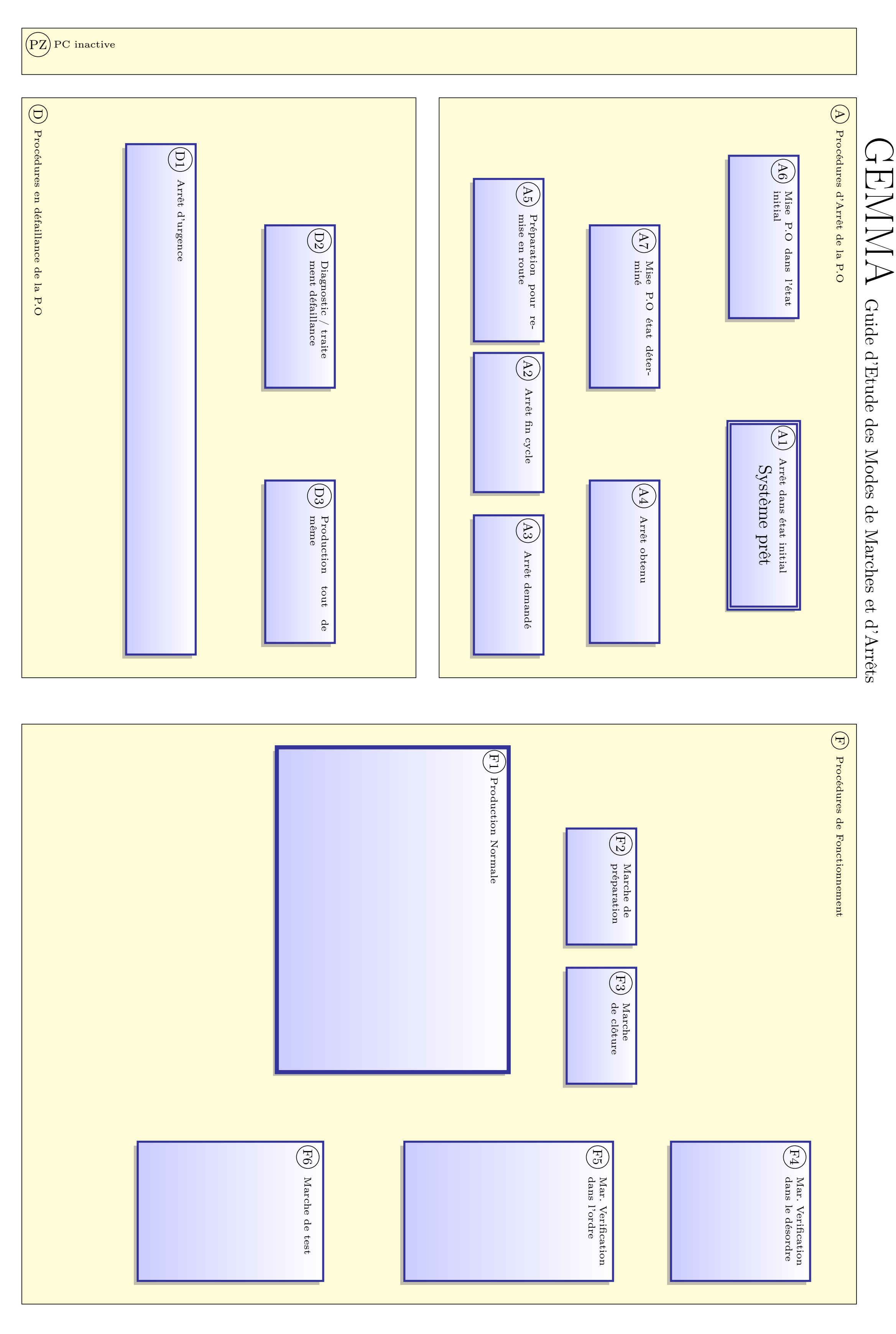


Figure 4.1: Le GEMMA

Exemple

Système de remplissage de bidons

Un système de remplissage de bidons est doté de 3 sous-systèmes :

- Un sous-système de remplissage non représenté et ne faisant pas partie de l'étude.
- Un sous-système de contrôle de la qualité de remplissage. Un bidon sur onze (tous les 2x5 en raison du poste de regroupement) est prélevé et aiguillé vers un poste de contrôle qui enverra alors ces bidons vers une autre unité de regroupement spécifique aux bidons contrôlés.
- Un sous-système permettant de regrouper les bidons afin de les conditionner par 2 pour la vente.

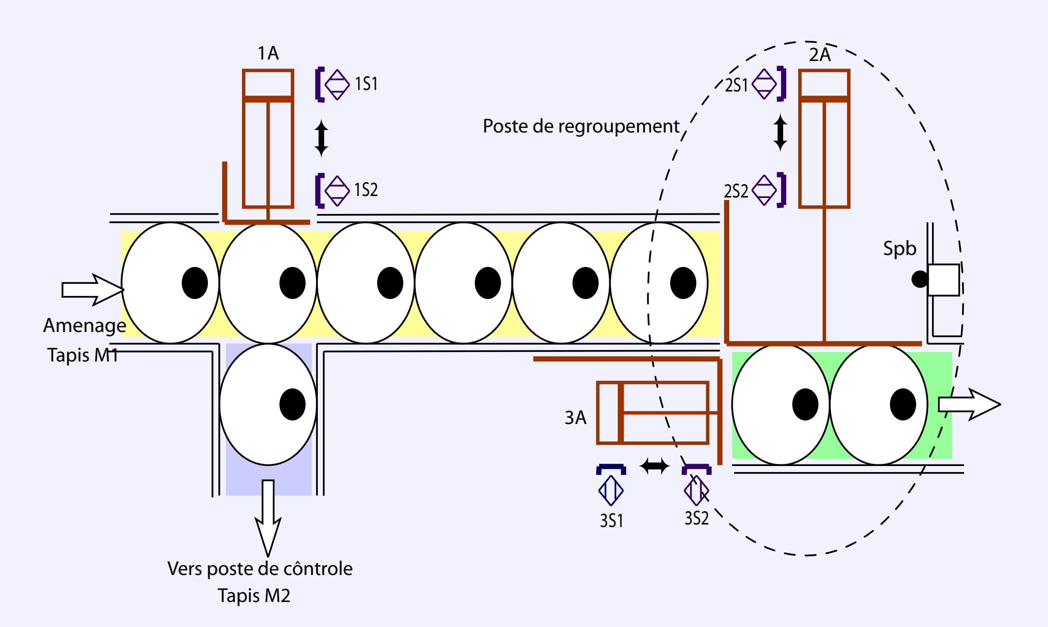


Figure 4.2: Système de remplissage de bidons

Description du système : A l'état initial, le vérin 2A est sorti, les vérins 1A et 3A sont rentrés. A la suite d'une autorisation de marche :

- Le tapis roulant est mis en mouvement par le moteur M1.
- Le vérin 3A nommé évacuateur, sort pour évacuer les bidons regroupés.
- Le vérin 2A nommé pousseur, rentre.
- Lorsque le détecteur Spb capte la présence d'un bidon, alors le tapis s'arrête. Le vérin 3A rentre puis le vérin 2A pousse 2 bidons.
- A la 5ème poussée de 2A, 1 bidon est prélevé pour contrôle en faisant sortir puis rentrer le vérin 1A dit préleveur. Le tapis M2 (mis en continu) dirige le bidon vers le poste de contrôle.

Modes des marches et d'arrêts Les modes de marches et d'arrêts sont décrits à l'aide du GEMMA joint figure (4.3).

- A la mise en énergie, l'initialisation de la partie opérative (PO) sera effectuée (si nécessaire) par un opérateur qui appuie sur le bouton poussoir "init".
- L'appui sur le bouton poussoir "dcy" lorsque le mode "Auto" est sélectionné, autorise le fonctionnement automatique du système. Pour quitter ce mode, l'opérateur doit appuyer sur le bouton poussoir "acy". la partie opérative termine son cycle et se retrouve arrêtée dans l'état initial.
- L'enclenchement du bouton d'arrêt d'urgence "Au" stoppe les actionneurs et met sous tension un voyant "Hau". Après désactivation de "Au", l'appui sur le bouton de réarmement "rer" initialise la PO conformément au GEMMA de la figure 4.3

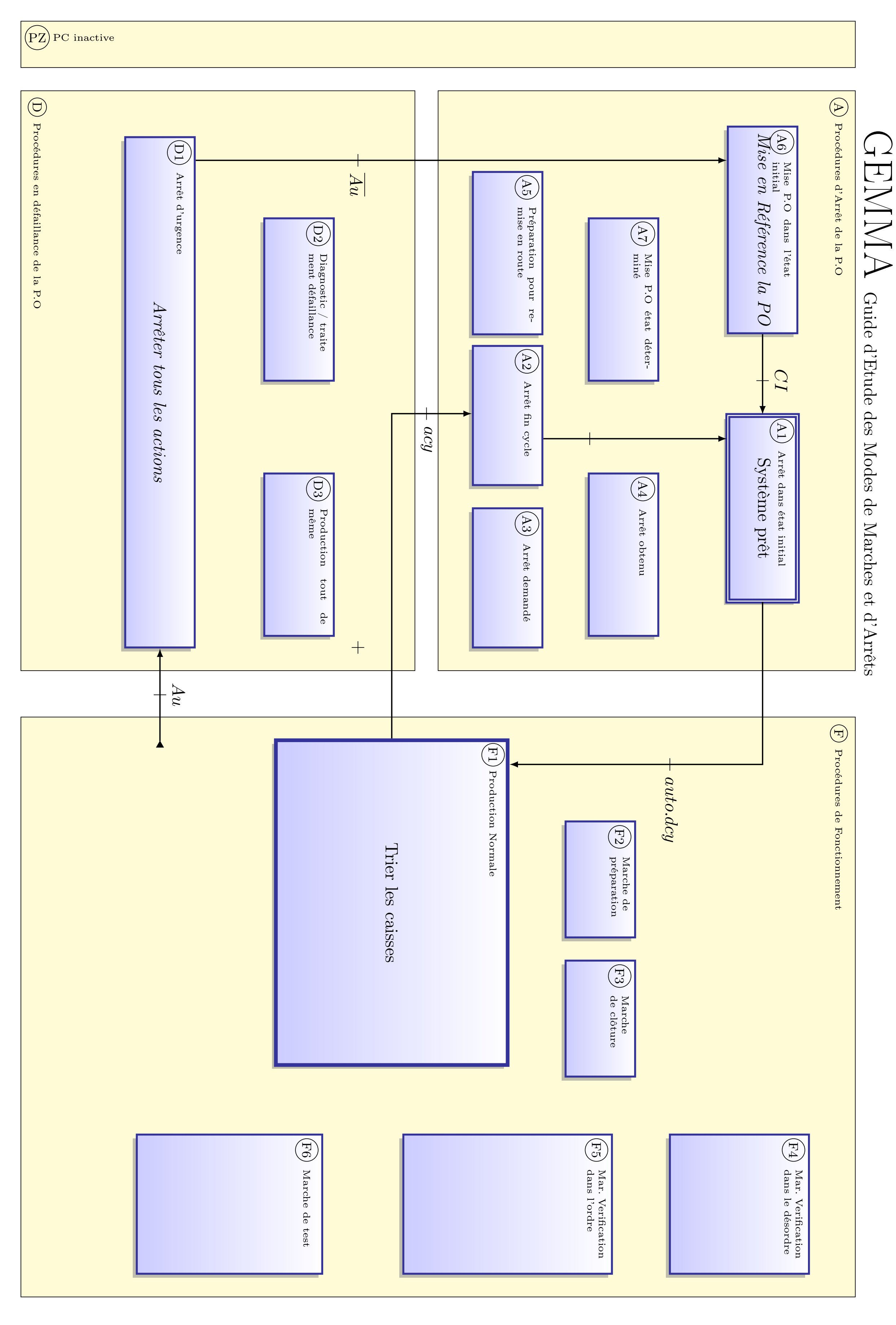


Figure 4.3: GEMMA : système de remplissage de bidons

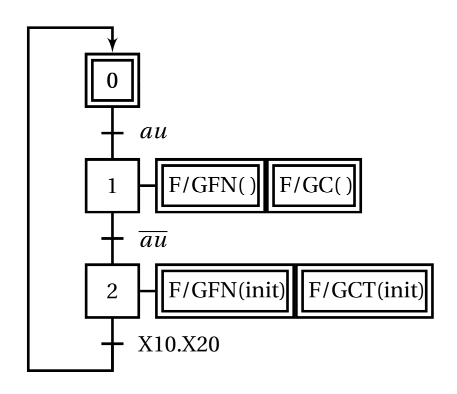


Figure 4.4: GRAFCET de surveillance GS

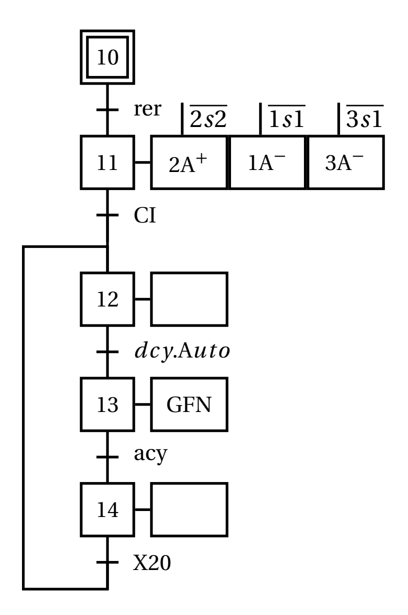


Figure 4.5: GRAFCET de conduite GC

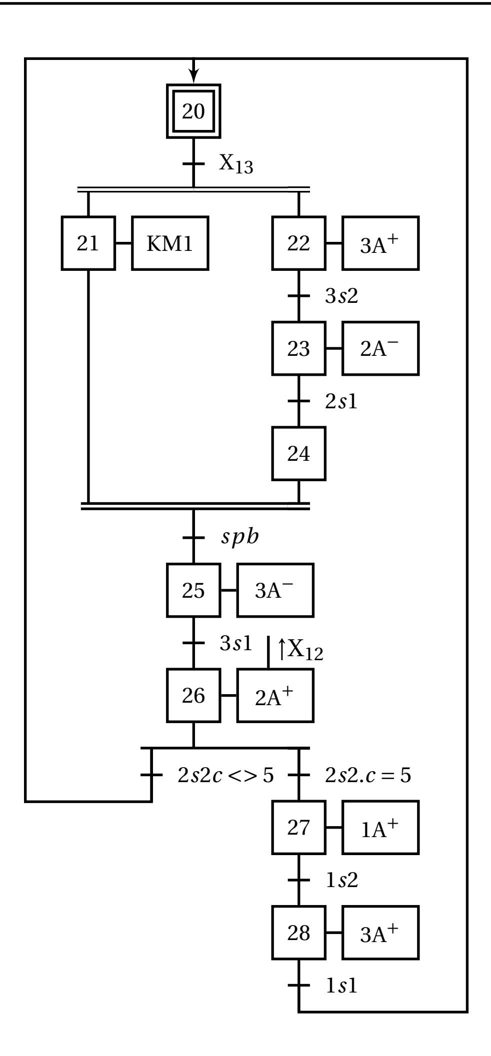


Figure 4.6: GRAFCET de Fonctionnement Normal GFN

Exercices

► Exercice 1

On considère deux chariots CH_1 et CH_2 pouvant ce déplacer sur deux rails de guidage, vers la gauche ou vers la droite entre deux positions a_1 et b_1 pour CH_1 et a_2 et b_2 pour CH_2 figure (4.7). Les chariots sont tous d'eux au repos (en a_1 et a_2), et lorsqu'on appuie sur bouton m, ils partent simultanément vers la droite. Le premier chariot qui arrive à l'extrémité droite repart à gauche et provoque le retour de l'autre chariot. Un nouveau départ ne peut être donné que s'ils sont tous d'eux en position gauche.

1. Écrire le GRAFCET correspondant au fonctionnement du système .

► Exercice 2

Reprendre les deux chariots de la figure (4.7). Lorsqu'un opérateur appuie sur le bouton m et si les deux chariots sont en a_1 et a_2, alors ceux-ci partent simultanément vers la droite. Un chariot CH_i qui arrive à droite au point b_i s'arrête. Le retour des deux chariots s'effectue soit dès que les deux chariots ont atteint b_1 et b_2, soit dès que CH_1 a atteint b_1 depuis 10 secondes. Dans ce cas, CH_2 repart vers la gauche sans avoir atteint b_2. Dans tous les cas, les deux chariots repartent simultanément vers la gauche.

1. Écrire le GRAFCET correspondant au fonctionnement du système .

► Exercice 3

Soient les deux chariots de la figure (4.7). Une cellule photoélectrique est positionnée en c. Lorsque le chariot CH_1 passe en ce point la cellule est obstruée.

Lorsqu'un opérateur appuie sur le bouton m, et si les deux chariots sont en a_1 et a_2, alors le chariot CH_1 part seul vers la droite. Son passage au point c autorise le départ du chariot CH_2 vers la droite. Si CH_1 arrive en b_1 avant que CH_2 n'arrive en b_2, alors il l'attend pendant 1 minute au plus. Passé ce délai le chariot CH_1 repart jusqu'en a_1. Le chariot CH_2 va jusqu'en b_2. Il repart vers la gauche jusqu'en a_2 après que le chariot CH_1 soit passé à nouveau au point c.

1. Écrire le GRAFCET correspondant au fonctionnement du système .

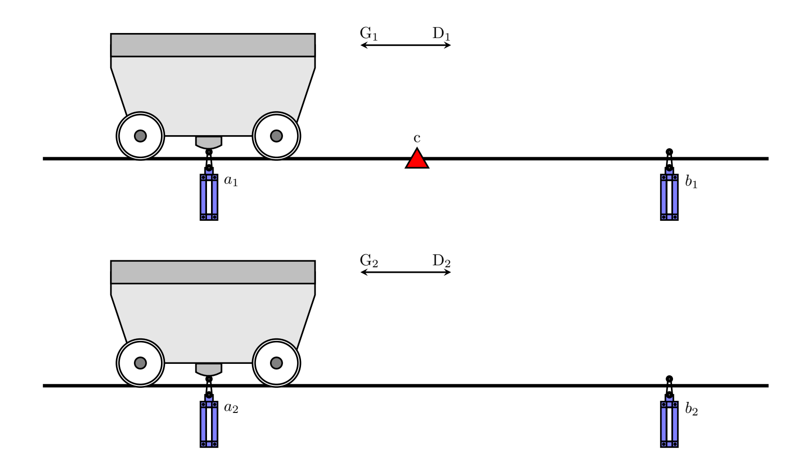
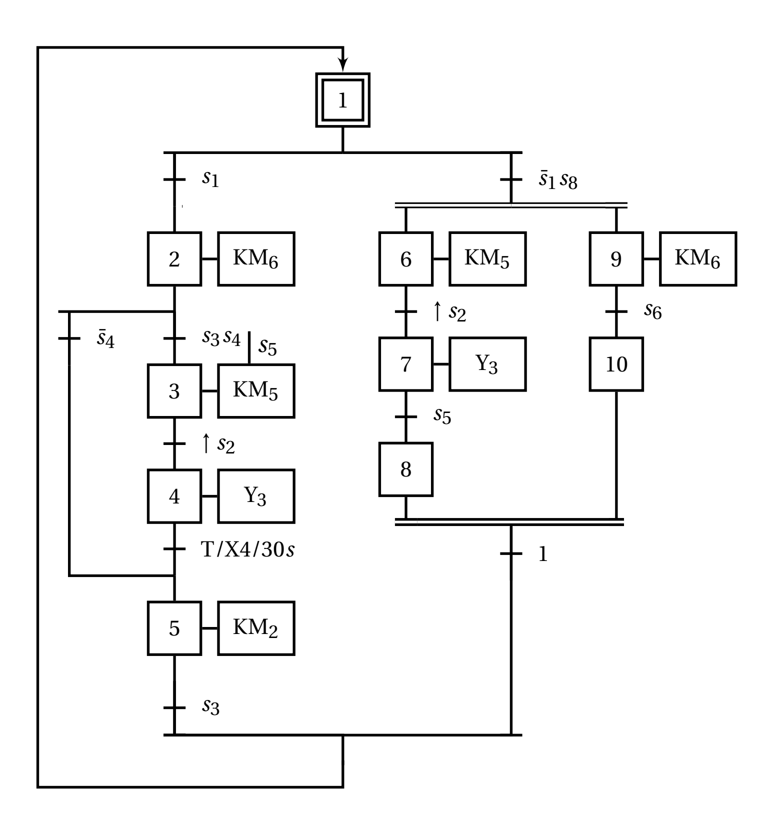


Figure 4.7: Système à deux chariots

▸ Exercice 4



1. Écrire le fonctions transitions puis établir leur schémas à contacts
2. Écrire les conditions d'activation et de désactivation de chaque étape en évolution synchrone et asynchrone
3. Avec une implantation en monostable puis en bistable, établir, en langage à contacts, le programme du GRAFCET.
4. L'automate utilisé possède 16 entrées dont les adresses mémoires sont désignées par la lettre "E" et 10 sorties dont les adresses mémoires sont désignées par "Q". les bits internes de la mé-

moire sont désignés par "M".

Dans un tableau, dresser les adresses des différentes variables utilisées.

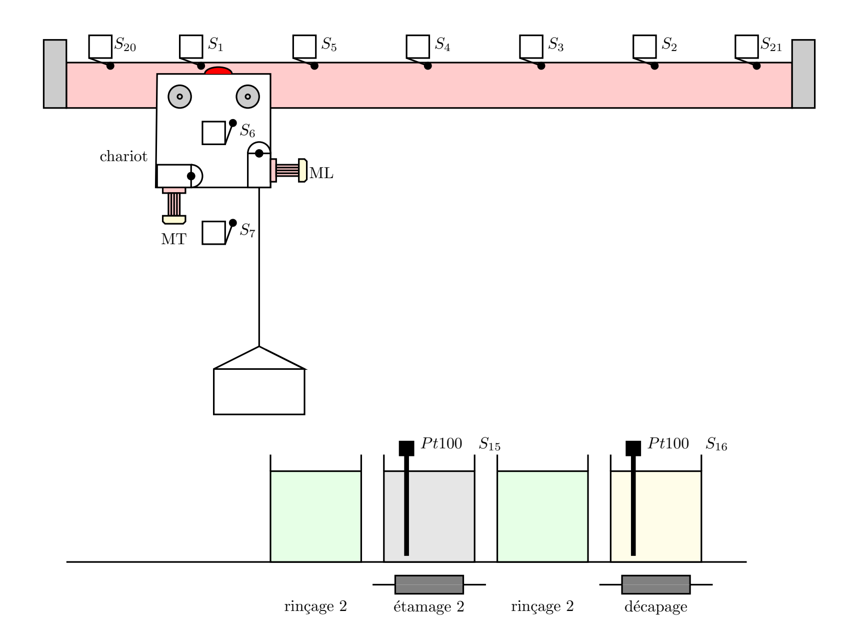
► **Exercice 5**

Système de traitement de surface

Soit une installation de traitement de surface comprenant un chariot auto-moteur desservant quatre bacs, un poste de chargement et de déchargement S_1. Des informations S_2 et S_5 permettent le positionnement au-dessus des différents postes. L'opérateur, après avoir accroché les pièces à traiter sur le cadre situé au point de chargement, en position basse, donne l'information de départ cycle par un bouton dcy.

- Le chariot auto-moteur élève le châssis en position haute.
- Il effectue une translation et se positionne au dessus de la première cuve en S_2.
- Il descend alors le châssis dans le bac.
- Il le laisse pendant le temps déterminé. A l'expiration du temps, le chariot remonte le châssis pour aller ensuite au bac suivant. Il recommencera les mêmes opérations jusqu'au poste de déchargement où l'opérateur décrochera les pièces avant de renvoyer le chariot. L'accrochage des tôles est exécuté manuellement.

Le temps de décapage, de rinçage et de l'étamage sont respectivement 20 secondes, 30 secondes et 3 minutes.



1. Écrire le GRAFCET du fonctionnement du système.
2. Établir les equations logiques en evolution synchrone et asynchrone.

► **Exercice 6**

La figure (4.8) représente un système de manutention qui comporte :

- Un chariot se déplaçant sur deux rails (action D vers la droite, G vers la gauche).
- Une pince pouvant descendre (action DP, fin quand fdp), prendre la pièce (action PP+, fin quand fpp+), monter (action MP, fin quand fmp) et libérer la pièce (action PP-, fin quand fpp-).
- Un plateau qui peut descendre (action V-, jusqu'à fv-) et remonter (action V+ fin quand fv+) après que le pousseur avait poussé la pièce.
- Un pousseur qui pousse la pièce (action P+, fin quand fp+) puis recule (action P-, fin quand fp-)

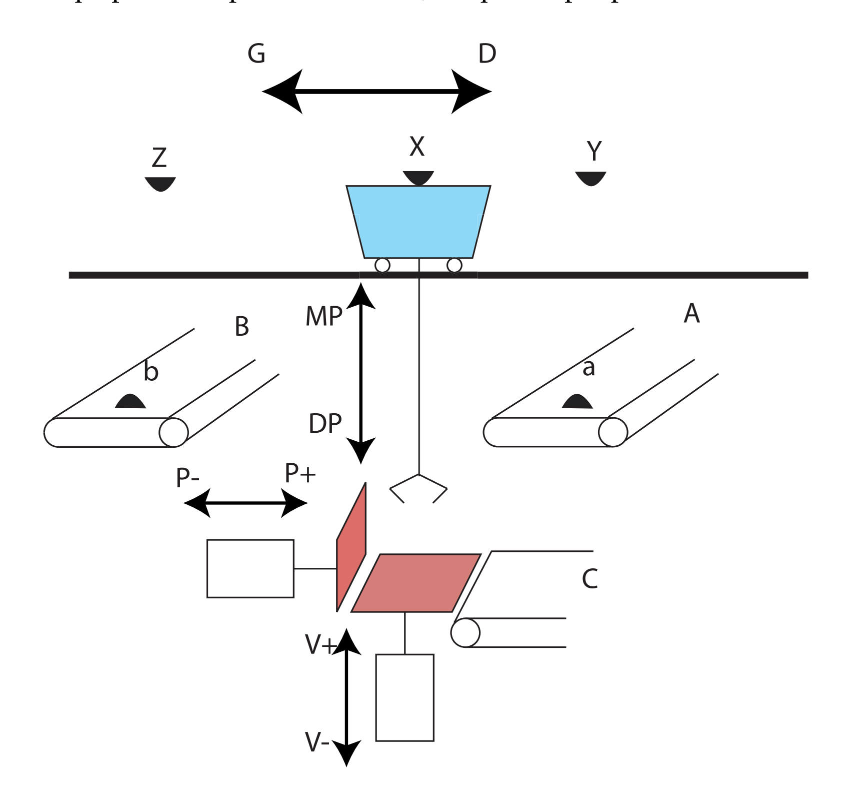


Figure 4.8: Système de manutention

Initialement, on suppose que la pince est en haut au dessus du plateau qui est aussi supposé en position haute. Les tapis A,B et C sont commandés par des systèmes non traités ici.

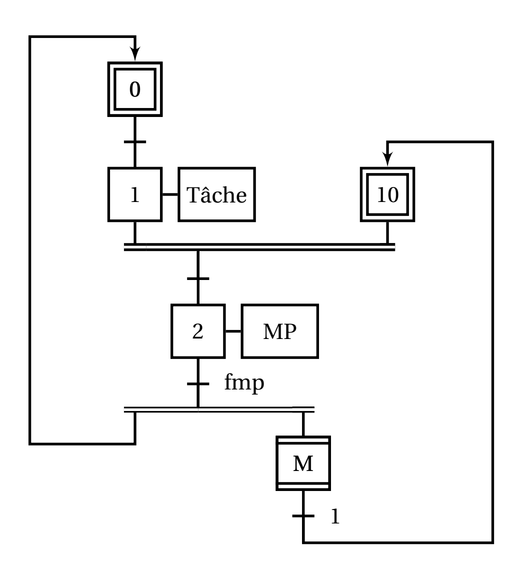


Figure 4.9: GRAFCET Principal

Le fonctionnement du système peut être décrit par le GRAFCET principal de la figure (4.9).

1. On suppose que seul le tapis A fonctionne. Si une pièce est présente en (a), le chariot se déplace pour la mettre sur le plateau. Compléter le GRAFCET principal puis écrire la tâche qui correspond à ce transfert de la pièce.
2. Écrire l'expansion de la macro M qui décrit l'évacuation de cette pièce (la mettre sur le tapis C)
3. Écrire les equations logiques de l'étape d'entrée et de sortie de la macro
4. Quel est le rôle de l'étape 10 ?
5. Le tapis B fonctionne, comment modifier la tâche précédente pour tenir compte de ce fonctionnement sachant que la prise d'une pièce sur A est prioritaire à la prise d'une autre en B.

► Exercice 7

Il est impératif lors de l'opération de conditionnement des feuilles d'aluminium, pour les protéger des éraflures, d'intercaler une feuille de cuir entre chacune des feuilles d'aluminium Figure (4.13). La consigne de marche (dcy) entraîne l'empilage des feuilles d'aluminium en intercalant des feuilles de cuir. Le cycle s'arrête dans le cas ou l'un des stocks est à zéro. Le cycle d'empilage débute toujours par la prise d'une feuille de cuir.

Les différents actionneurs du système ainsi que les capteurs utilisés sont indiqués dans les tableaux suivants :

Actionneurs

DESIGNATIONS	COMMANDES
Vérin double effet A associé à un distributeur 5/2 bistable (24V)	A+ (Sortir le vérin), A - (Rentrer le vérin)
Vérin double effet B associé à un distributeur 5/2 bistable (24V)	B+ (Sortir le vérin), B- (Rentrer le vérin)
Vérin double effet C associé à un distributeur 4/2 monostable (24V)	C (Sortir le vérin)
Générateur de vide venturi V associé à un distributeur 3/2 monostable (24V)	V (Créer une aspiration)

Capteurs

DESIGNATION	FONCTION	TYPE
a_0	Vérin A rentré	Détecteur magnétique
a_1	Vérin A sorti	Détecteur magnétique
b_0	Vérin B rentré	Détecteur magnétique
b_1	Vérin B sorti	Détecteur magnétique
c_0	Vérin C rentré	Détecteur magnétique
c_1	Vérin C sorti	
j	Présence de vide dans la ventouse	Vacuostat
cu	Présence d'une feuille de cuir	Détecteur de proximité
al	Présence d'une feuille d'aluminium	Détecteur de proximité

Modes des marches et d'arrêts

Les modes de marche à considérer sont les suivants :

- marche automatique,
- initialisation automatique de la partie opérative,
- marche manuelle : des boutons de commande manuelle doivent permettre de démarrer le cycle aller/retour de chaque vérin et d'assurer le vide,
- arrêt d'urgence.

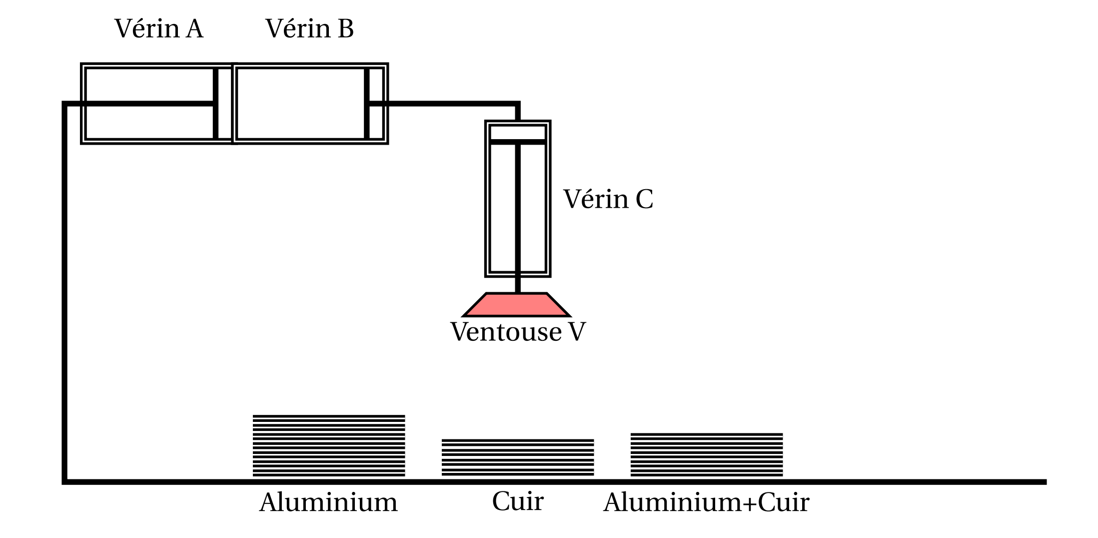


Figure 4.10: Machine à empiler

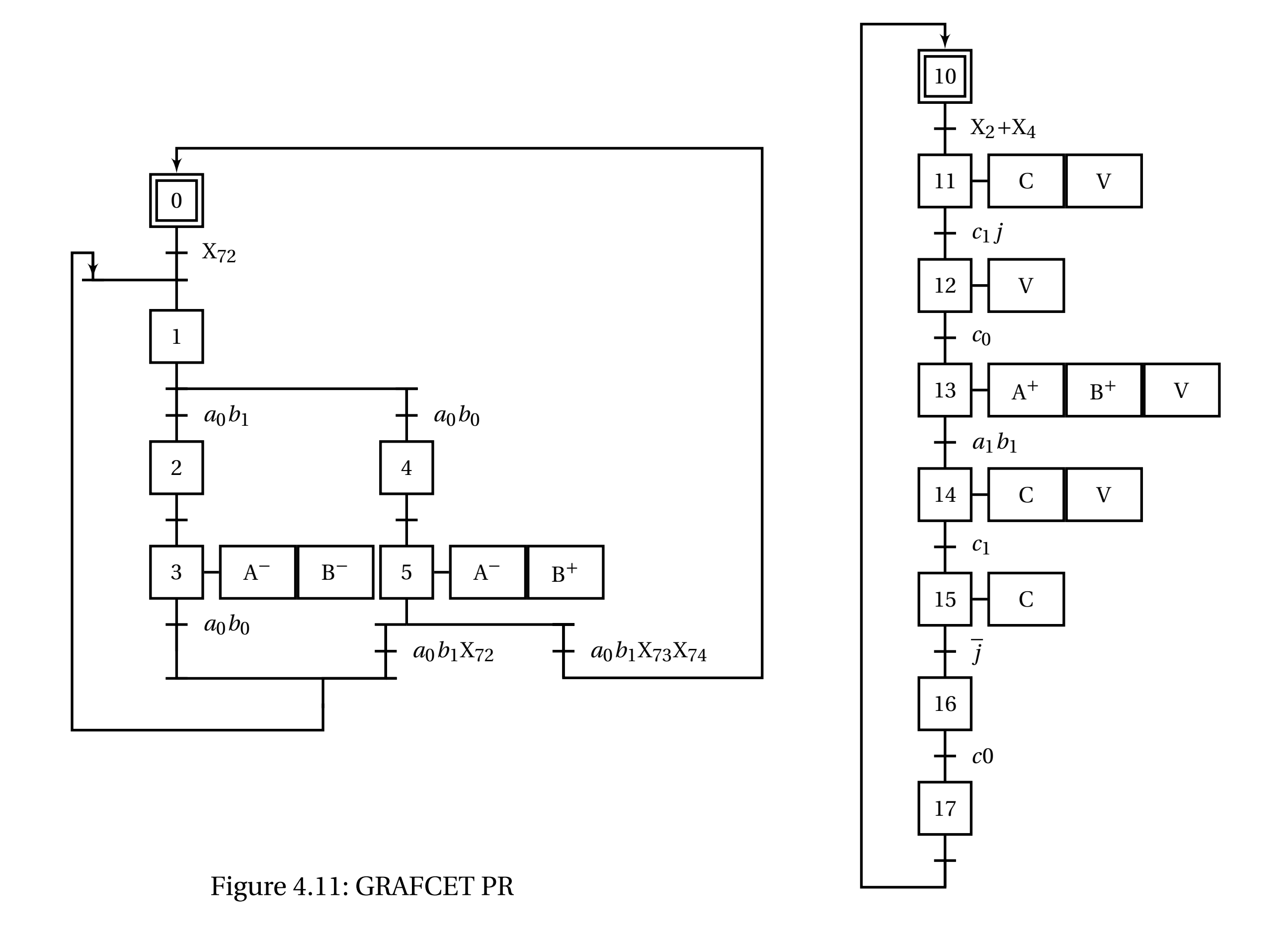


Figure 4.11: GRAFCET PR

Figure 4.12: GRAFCET GT

On peut décrire tout d'abord les modes de marche souhaité par le GEMMA de la figure (4.15). Les GRAFCETS de sécurité (GS) et de conduite (GC), sont donnés par les figures (4.13)-(4.14).

Les GRAFCETS de production (PR) et de tâche (GT), de point de vue partie opérative, sont donnés par les figures (4.11)-(4.12)

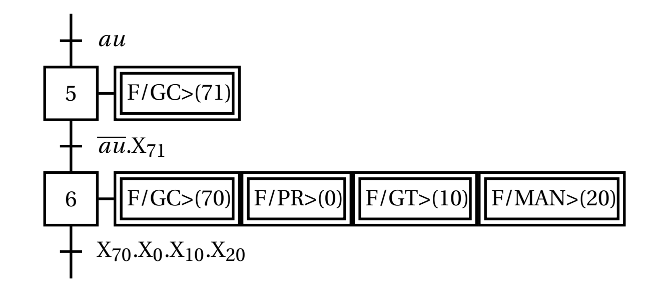


Figure 4.13: GRAFCET de Sécurité (GS)

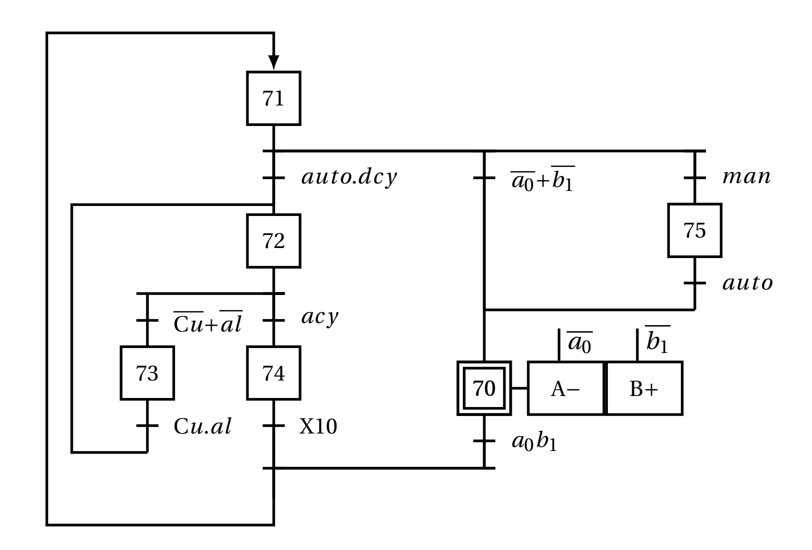


Figure 4.14: GRAFCET de Conduite GC

Etude du cycle de production et programmation de l'API

1. Dans le GRAFCET tâche GT, on peut lire la réceptivité X_2+X_4. Quel est son rôle ?
2. Proposer une réceptivité de la transition 17.
3. Proposer une réceptivité de les transition 2 et 4 du GRAFCET PR
4. Quel est le rôle de l'étape 16 du GRAFCET GT ?
5. Quel est le rôle de l'étape 17 du GRAFCET GT ?
6. Pourquoi l'action V se répète dans les étapes 11-12-13-14 ? Proposer une autre solution. (ne représenter que la partie à modifier).
7. Représenter les étapes 11-12-13-14-15 par une macro étape M.
8. Établir les fonctions transitions avec leur schémas à contacts et le programme automate de l'expansion de la macro M en utilisant les mémoires (RS) en évolution synchrone.

Etude des modes de marches et d'arrêts

- Lors du fonctionnement normal du système en mode automatique, quelle est l'étape active des GRAFCET (GC) ?
- A quoi correspond l'étape 73 du GRAFCET (GC) ?
- A quoi correspond l'étape 74 du GRAFCET (GC) ?
- A quoi correspond l'étape 70 du GRAFCET (GC) ?
- On désire ajouter le mode manuel au système en actionnant le sélecteur de modes "man". A partir de l'état A1, représenter les rectangles d'états ainsi que les liens qui correspondent à ce mode tout en tenant compte éventuellement de la mise en référence de la P.O en sortant de ce mode.
- Établir le GRAFCET correspondant à ce mode (étape initiale numérotée 20).

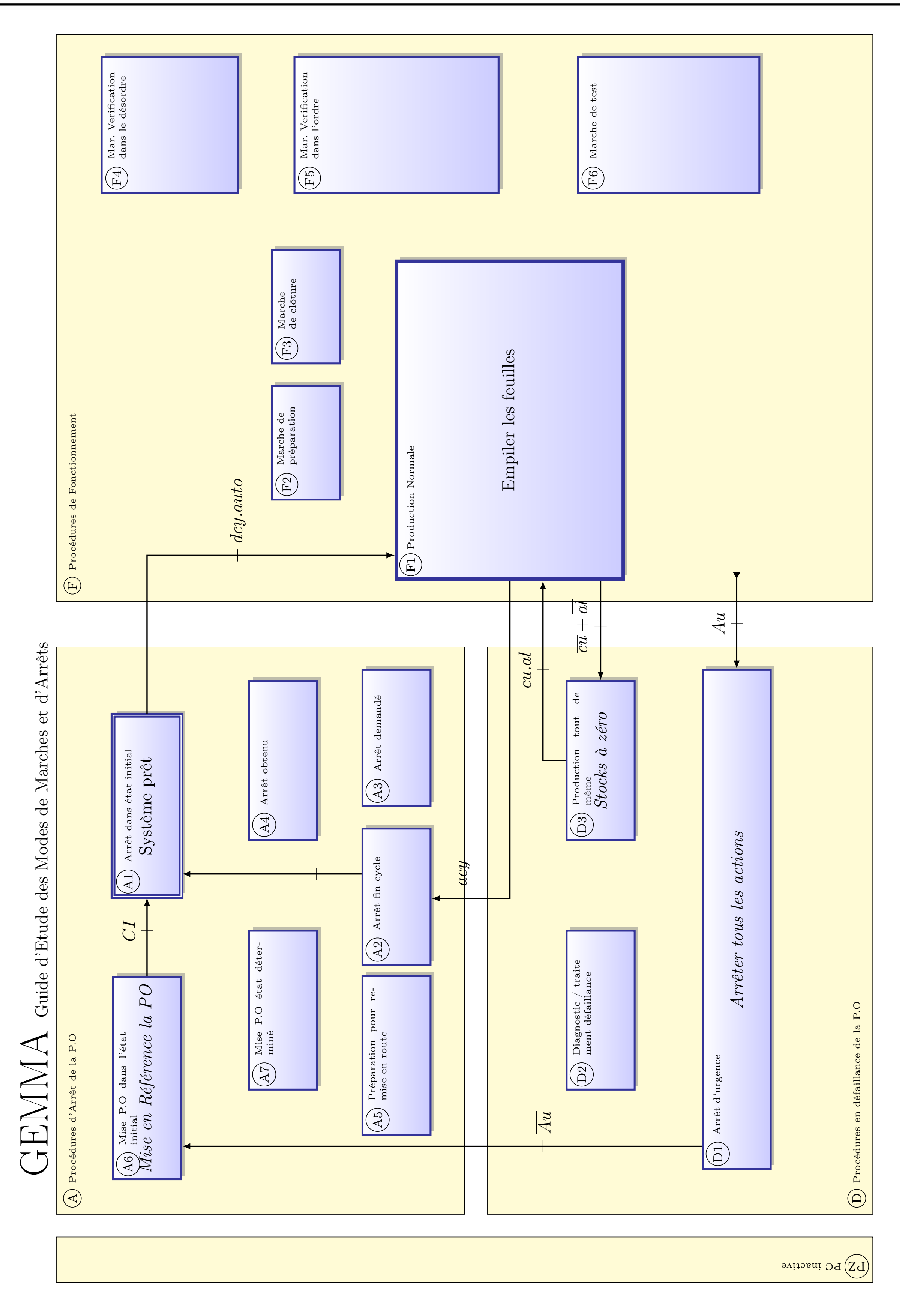


Figure 4.15: GEMMA : Machine à empiler

Bibliographie

[1] Michel BERTRAND. Automates programmables industriels. Techniques de l'Ingénieur, traité Informatique industrielle S8015.

[2] Patricia JARGOT. Langages de programmation pour API. Norme IEC 1131-3. Techniques de l'Ingénieur, traité Informatique industrielle S8030.

[3] S. Moreno et E. Peulot. Le GEMMA, Collection A. Capilez. Educalivre 1997.

[4] S. Moreno et E. Peulot. Le GRAFCET conception et implentation dans les automates programmables i dustriels, Collection A. Capilez. Educalivre 1997.

[5] Réné David et Hassane Alla. Du GRAFCET aus réseaux de Petri, Traité des Nouvelles Techniques série Automatique. HERMES 1992.

[6] Jean-Yves FABERT. Automatismes et automatique. Ellipses 2005.

Table des figures

Liste des tableaux